

# ON THE COMPRESSION OF MRI CROSS-SECTIONS

by

R. Kalyanaraman



DEPARTMENT OF MATHEMATICS

**INDIAN INSTITUTE OF TECHNOLOGY KANPUR**

JUNE, 1999

# ON THE COMPRESSION OF MRI CROSS-SECTIONS

*A Thesis Submitted*  
in Partial Fulfillment of the Requirements  
for the Degree of

**DOCTOR OF PHILOSOPHY**

*by*

**R. KALYANARAMAN**

*to the*

**DEPARTMENT OF MATHEMATICS**

**INDIAN INSTITUTE OF TECHNOLOGY KANPUR**

**June, 1999**

24 JUN 2002 / Math

पुरुषोत्तम काशीनाथ केलकर पुस्तकालय  
भारतीय प्रौद्योगिकी संस्थान कानपुर  
अवधि क्र० A.....139687.....



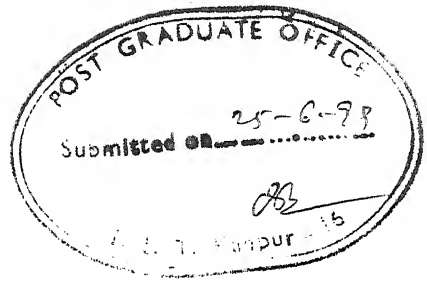
A139687

***DEDICATED***

***TO***

***MY PARENTS AND TEACHERS***





### CERTIFICATE

It is certified that the work contained in the thesis entitled, "**ON THE COMPRESSION OF MRI CROSS-SECTIONS**", by R. Kalyanaraman, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

R.K.S. Rathore

Professor

Department of Mathematics

I. I. T. Kanpur

June, 1999

## ACKNOWLEDGEMENTS

I have great pleasure in expressing my deep sense of gratitude to my thesis supervisor, **Prof. R.K.S. Rathore**, for his excellent guidance, inspiring discussions and unfailing help at every stage of my research. I am indebted to Dr. (Mrs.) Rathore for her tolerance during long discussions with my *Guruji* Prof. R.K.S. Rathore.

I am very much grateful to Prof. Dr. R.K. Gupta, Department of Radiology, Sanjay Gandhi Post Graduate Institute of Medical Sciences, Lucknow for his valuable guidance, encouragement and providing us with very useful and valuable MRI data to carry out this research.

I am thankful to Prof. P.C. Das, Prof. G.P. Kapoor, Prof. SobhaMadan, Prof. M.R.Sridharan, Prof. B.V. Rathishkumar and Prof. P. Munshi for their help and encouragement.

I am highly indebted to my friends Mahesh, S.B.Rao and V.S.N Kaliprasad for the kind encouragement and help during the hours of need. I am also thankful to Sushmita, Divya and Satya

I will always cherish the company of my friends Parasar Mohanty, Behra, Alok, Manish, Balaji, Siva, Rajesh, Bala, Senthil Aandavar, Senthil, Jeyaprakash Sridevi, Anuradha, Srini, Pachamuthu, Nagendran, GaneshBabu, Ramanujam and Jeyakumar.

I can't find words to express the intensity of my feelings for the love and affection, I got from my Amma, Appa, Anna and Manni. With deep sense of gratitude, I wish to acknowledge my gratefulness to my chithappa Dr.S.Jayaraman, who inspired me and sculpted my research preferences. I am indebted to my chithappa Sri.S.Krishnamoorthy for his help, and encouragement. I am grateful to all my chithis, chithappas and cousins for their kind encouragement, love and affection I got from them during all these years.

My special thanks to Viji didi, Prabhakar Akhila for all their help and affection. I take this opportunity to convey my regards to my Teakwando Sir, Mr.Balram Yadav.

I am grateful to my friends Uma and Ananth who unceasingly sent their love from oceans away. I am also thankful to Shankar, Srikanth, Gomathi Sankar, R.Ramesh, Babu, Nataraj, Sriram sundar, Srinivasan and all other friends for their constant encouragement.

Finally, I would like to acknowledge the financial support given by C.S.I.R, India for carrying out my research.

R.Kalyanaraman.

# CONTENTS

	PAGE
LIST OF IMAGES	vii
LIST OF GRAPHS	viii
LIST OF TABLES	ix
 CHAPTER – I	
INTRODUCTION	1
1.1 Digital Image Model	2
1.2 Variable Length Coding	4
1.2.1 Huffman Coding	5
1.2.2 Arithmetic Coding	7
1.2.3 Dictionary based Compression	9
1.3 Lossy Compression	11
1.3.1 Predictive Coding	12
1.3.2 Transform Coding	12
1.3.3 Quantization	15
1.4 Standard Compression Methods	16
1.4.1 JPEG Standard	16
1.4.2 Graphics Interchange Format	18
1.4.3 Fractal Compression	18
1.4.4 Wavelet based Compression	19
1.5 MRI image Compression	21
1.5.1 Magnetic Resonance Imaging - Basics	22
1.6 Experimental Setup	30
1.7 Thesis Organization	31
 CHAPTER - II	
COMPRESSION THROUGH TRIGONOMETRIC POLYNOMIALS	39
2.0 Introduction	39
2.1 Besov Spaces	40
2.2 Equivalent Norms	45
2.3 Hardy Type inequality and bound on Fourier coefficients	47
2.4 Direct and Inverse Theorems	50
2.5 Compression Algorithm and Storage bounds	57
2.6 Compression of Raw data	61
2.8 Experimental Results	62
 CHAPTER-III	
COMPRESSION THROUGH ALGEBRAIC INTERPOLATION	76

3.0	Introduction	76
3.1	Storage of Finite Differences	78
3.2	Bernstein's Lemma and certain estimates for smooth images	88
3.4	Direct and Inverse Theorems	93
3.5	Experimental Results	98
CHAPTER-4	COMPRESSION THROUGH REVERSIBLE SMOOTHING	104
4.0	Introduction	104
4.1	Approximation Processes	106
4.2	Compression Algorithm	110
4.2.1	Compression	110
4.2.2	Decompression	110
4.3	Non Redundant Sampling	123
4.4	Wavelet Basics	127
4.4.1	Compression Algorithm and storage bounds	129
4.4.2	Experiments	130
REFERENCES		136

## List of Images

Image No.		Page
1.5.1	Raw data	30
1.5.2	Image	30
1.5.3	Magnitude reconstructions – Uncompressed (Test Image Set 1)	34
1.5.4	Sharpened Images ( Test images Set 2)	35
1.5.5	PD T1 and T2 Weighted Images (Test images Set 3)	36
2.7.1	55 % Compression using Algorithm 2.5.1	64
2.7.2	60 % Compression using Algorithm 2.5.1	65
2.7.3	65 % Compression using Algorithm 2.5.1	66
2.7.4	T1, T2 and Spin Density Computed from Uncompressed images	69
2.7.5	Computed from 47.38 % compressed image set	70
2.7.6	Computed from 57.68 % compressed image set	71
2.7.7	Computed from 66.75 % compressed image set	72
2.7.8	Compression of Raw data	73
2.7.9	Compression of Raw data	74
3.5.1	Non-lossy compression through Algorithm 3.2.1 Images after decompression	99
3.5.2	Image compression through JPEG	100
3.5.3	Compression of Sharpened Images	101
4.0.1	Uncompressed Image	105
4.0.2	Smooth Image	105
4.0.3	Decompressed Image	105
4.0.4	Error Image	105
4.2.1	Poisson process – Images after decompression	118
4.2.2	Fejer Smoothing – Images after decompression for various values of N	120
4.2.3	Fejer Smoothing – Images after decompression for various values of N	120
4.2.4	Compression of Sharpened images	122
4.2.5	Steklov Smoothing	123
4.4.1	Compression through Wavelets	133
4.4.2	Compression through Wavelets	134

## List of Graphs

Graph No.		Page
2.6.1	Decay of Fourier Coefficients	62
2.7.1	Percentage Compression Vs Relative error – img60	67
2.7.2	Percentage Compression Vs Relative error – img64	67
2.7.3	Percentage Compression Vs Relative error – img68	68
2.7.4	Percentage Compression Vs Relative error – img72	68
2.7.5	Percentage Compression Vs Average 11-error (Compression of Raw Data)	74
3.0.1	Histogram of the image	77
3.0.2	Histogram of the first difference	77
3.2.1	Graph of $\delta$ Vs $\omega_1(f, \delta)_1$	83
3.2.2	Graph of $\delta$ Vs $\omega_1(f, \delta)_2$	83
3.2.3	Graph of $\log(\delta)$ Vs $\log(\omega_1(f, \delta)_1)$	84
3.2.4	Graph of $\log(\delta)$ Vs $\log(\omega_2(f, \delta)_1)$	84
3.2.5	Graph of $\log(\delta)$ Vs $\log(\omega_1(f, \delta)_2)$	85
3.2.6	Graph of $\log(\delta)$ Vs $\log(\omega_2(f, \delta)_2)$	85
3.3.1	Graph of $\delta^2$ Vs $\omega_2(f, \delta)$	92
3.3.2	Graph of $\delta^3$ Vs $\omega_3(f, \delta)$	92
4.2.1	Windows	111
4.2.2	Percentage Compression Vs Relative Error (Poisson Process) – img60	113
4.2.3	Percentage Compression Vs Relative Error (Poisson Process) – img64	113
4.2.4	Percentage Compression Vs Relative Error (Poisson Process) – img68	114
4.2.5	Percentage Compression Vs Relative Error (Poisson Process) – img72	114
4.2.6	Variation of gain with respect to smoothness - grl1	115
4.2.7	Variation of gain with respect to smoothness - grl2	115
4.2.8	Variation of gain with respect to smoothness - gal1	116
4.2.9	Variation of gain with respect to smoothness - gal2	116
4.2.10	Variation of Smoothness with N in Fejer's smoothing - all	119
4.4.1	Percentage Compression Vs Relative error – Daub 4	131
4.4.2	Percentage Compression Vs Relative error – Daub 12	132
4.4.3	Percentage Compression Vs Relative error – Daub 20	132

## List of Tables

Table No.		Page
1.5.1	Compression of test image set 1 by standard methods	37
1.5.2	Compression of test image set 2 by standard methods	38
1.5.3	Compression of test image set 3 by standard methods	38
2.7.1	Effect of Compression in the computation of Tissue Parameters	73
3.2.1	$\alpha$ - Estimates for test image set 1 - 11	86
3.2.2	$\alpha$ - Estimates for test image set 1 – 12	86



# CHAPTER I: INTRODUCTION

The main objective of image data compression is to reduce the volume of data needed in order to represent an image. The major motivation for the study of compression, arises due to the restricted memory resources for data storage. The amount of data involved in Picture Archiving and Communication Systems (PACS) [1], is a typical example where data compression schemes can have a great economic impact. The advent of Internet and networking in general, has vastly increased the information transfer. The amount of data transmitted is limited by the available channel bandwidth. Data compression plays a major role not only in storage but also in transmission. Another useful application of data compression is the development of fast algorithms where the operations required to implement an algorithm is reduced by working with the compressed data.

Data compression can be achieved by exploiting the redundancy, which is a characteristic, related to predictability, correlation etc., in the data. The two main components of data compression are **redundancy modeling** and **coding**. In the first step (redundancy modeling), the data is transformed into a suitable domain where maximum information is packed in minimum number of samples. The objective of that transformation is to extract the redundancy present in the original data. Such a transformation will be data dependent, and will be based on the redundancy model. The second step (coding) actually stores the new data form with some predefined rules and conventions so as to obtain more compact data storage.

In digital computers, data is represented as a string of binary digits, popularly known as bits. A bit can represent two distinct states 0 and 1. A string of  $k$  bits can be used to represent  $2^k$  distinct states. Code is a string of bits assigned to each symbol of the data. Symbols in the data, which could be letters, signs, numbers and marks etc., are mapped to

codes and these codes are stored in digital computers. The original data can be recovered by using the relation between the symbols and codes.

Let  $S = \{ s_1, \dots, s_m \}$  be a set of  $m$  distinct symbols or states. A symbol is an atomic unit of information. The minimum number of bits used to code each symbol in  $S$  is the least positive integer  $k$  such that  $2^k \geq m$ . If the data set  $D = \{ d_1, \dots, d_n \}$  of length  $n$ , then it can be coded using  $nk$  bits. This kind of coding is referred to as Fixed Length Coding (FLC), where the code length of each symbol in the data sequence is constant.

Accurate model of the data redundancy, helps to characterize data sets based on the considered redundancy. Compression algorithms based on the redundancy models are very effective for the class of data characterized by certain measure of redundancy. The objective of this work is centered on the compression of image data, particularly for the images arising from Magnetic Resonance Imaging (MRI). Medical imaging in general aims at producing images of sections of a physical object for diagnostic purposes. MRI is one of the most prominent medical imaging modality. MRI images have better soft tissue contrast than the images produced using Computerized Tomography (Hermen, [27]). Another important area where data compression schemes are highly applicable is the Teleradiology (Gitlin, [22]), or the practice of radiology at a distance. It involves transmission of medical images over a distance. Not much loss of information due to compression and decompression can be tolerated in most medical applications. These stringent requirements are the motivation for designing sophisticated compression algorithms for medical images.

## 1.1 Digital Image Model:

An image in general is an analog intensity distribution function which can be viewed as a mapping from  $[0, 2\pi] \times [0, 2\pi]$  to  $[0, 1]$ . This continuous intensity distribution function is digitized for storing the image in a digital computer. Let  $f$  be the analog intensity distribution function given by,

$$f: [0,2\pi]^2 \rightarrow [0,1]$$

Consider for  $0 \leq i \leq N, 0 \leq j \leq N$ ,

$$X_{ij}(x,y) = \begin{cases} N^2 & \text{if } \frac{i}{N} \leq x < \frac{i+1}{N}, \frac{j}{N} \leq y < \frac{j+1}{N} \\ 0 & \text{else} \end{cases}$$

Let,

$$P_{ij}(f) = \iint_I f(x,y) X_{ij}(x,y) dx dy, \text{ where } I = [0,2\pi]^2$$

$$\text{and } f \approx F(x,y) = \frac{1}{N^2} \sum_{i,j}^N P_{ij}(f) X_{ij}(x,y). \quad (1.1.1)$$

Let  $F$  be the digital image.  $P_{ij}$  's are the pixel values to be stored for representing the digital image.

### Depth and Resolution of a digital image:

Resolution of a digital image is the discretization of the domain  $[0,2\pi] \times [0,2\pi]$ . If  $N \times N$  is the discretization of the domain where  $N = 2^m$  for some  $m \in \mathbb{Z}_+$ , then the resolution of the digital image is  $m$ . Cases of the most practical interest are  $m = 7, 8, 9, 10$ .

Depth is the discretization of the range  $[0,1]$  and it is also the number of distinct gray levels the image can take. It is expressed in bits per pixel (bpp). An image is of  $b$ -depth if the range  $[0,1]$  is discretized into  $2^b$  parts.

Resolution and depth are the two quantities that determine the storage requirements. Throughout this thesis, by data we mean image data. The depth of the image would determine the number of states or symbols an image can take. Some useful relationship between the depth and resolution of a digital image is obtained in the fourth chapter of the present thesis.

## 1.2 Variable Length Coding (VLC):

In 1952, Huffman [30] gave a coding scheme where the code length is not constant for each symbol. It is based on the idea that the frequently occurring symbols can be coded with codes of smaller length as compared with the less frequently occurring symbols. The success of this coding in terms of compression is based on the hypothesis that all symbols in a data sequence are not equally likely. It is necessary for such codes to be uniquely decodable. Before going into the details of VLC, let us introduce some standard definitions, which will be used frequently.

Frequency distribution of the symbol set  $S$  consisting of  $m$  symbols, in the data  $D$  of length  $n$ , is the collection of positive numbers,  $Fr = \{f_1, \dots, f_m\}$  one for each state, defined by  $f_i = \#\{d_k \in D : d_k = s_i\}$ , where  $\#$  denotes the cardinality of the set. It follows that  $\sum_{i=1}^m f_i = n$ . In 1940 's, Shanon [68] defined the zero order entropy of a data sequence  $D$  to be

$$e = e(D) = -\sum_{i=1}^m p_i \log_2 p_i, \quad (1.2.1)$$

where  $p_i$  is the probability of occurrence of  $i$ th symbol in the data sequence  $D$ . Let  $C = \{c_1, \dots, c_m\}$  be a code book. It follows that the average code length  $C$ , to represent the data sequence  $D$  is  $e_c(D)$  given by,

$$e_c(D) = \sum_{i=1}^m p_i |c_i|, \text{ where } |c_i| \text{ stands for the length of the code } c_i.$$

### Theorem 1.2.1 (Shanon Entropy Theorem):

Given a data sequence of symbols, for any code book given to that symbol set, the average code length using the code book to represent the data sequence is always greater than or equal to the zero order entropy of the data sequence.

i.e. 
$$e_c(D) = \sum_{i=1}^m p_i |c_i| \geq e(D) = -\sum_{i=1}^m p_i \log_2 p_i.$$

We state the definition of Markov process, before describing certain algorithms. These algorithms try to achieve minimum code length prescribed by Shanon [67].

Let  $\{d_n\}$  be a data sequence generated in some process. It follows a  $k$ -th order Markov model (see, Sayood, [65]) if the knowledge of past  $k$ -symbols is equivalent to the knowledge of the past history of the process.

$$P(d_n | d_{n-1}, \dots, d_{n-k}) = P(d_n | d_{n-1}, \dots, d_{n-k}, \dots).$$

The values taken by  $(d_{n-1}, \dots, d_{n-k})$  are referred to as the states of the process. For an image of depth  $b$ , the number of states the pixel values can take is  $2^b$ . In the sequel we will use  $2^b$  as the number of elements in the symbol set  $S$ .

### 1.2.1 Huffman Coding:

Huffman [30] proposed a method for reducing the code length for data sets in which each symbol occurs with a different probability and has given an algorithm for constructing variable length codes. We present the algorithm here.

Consider, a zero order Markov process, given by a symbol set,  $S = \{s_1, \dots, s_{2^b}\}$ . In the case of images, elements of the symbol set  $S$  will contain gray level values. Here  $S$  denotes set of states an image of  $b$  bpp depth can take. And  $Fr$  is the frequency set given by  $Fr = \{f_1, \dots, f_{2^b}\}$ , where  $f_i$  denotes the number of occurrences of the symbol  $s_i$ . Given the data set this quantity is calculable. This is a redundant information present along with the data.

### Coding:

**Step 1:** We pick two symbols from the set  $S$  which have frequencies lower than the other symbols in the given data set  $D$ . Let the chosen symbols be denoted by  $c$  and  $d$ .

Let  $\sigma = \sigma^{2^b} : \{1, 2, \dots, 2^b\} \rightarrow \{1, 2, \dots, 2^b\}$  denote a permutation such that

$c = s_{\sigma(2^b-1)}, d = s_{\sigma(2^b)}$ . Then we define a new Markov source consisting of the set of symbols,  $S^{2^b-1} = \{s_{\sigma(1)}, \dots, s_{\sigma(2^b-2)}, t^{2^b-1}\}$  and the set of frequency distribution,  $Fr^{2^b-1} = \{f_{\sigma(1)}, \dots, f_{\sigma(2^b-2)}, f_{\sigma(2^b-1)} + f_{\sigma(2^b)}\}$ . The  $2^b$  state problem has reduced to a  $2^{b-1}$  states problem of finding a code book  $C^{2^b-1} = \{c_1^{2^b-1}, \dots, c_{2^{b-1}}^{2^b-1}\}$  for  $S^{2^b-1}$ . In summary, the first step can be described as

$$\{S, Fr, C\} \Rightarrow \sigma^{2^b} \Rightarrow \{S^{2^b-1}, Fr^{2^b-1}, C^{2^b-1}\}.$$

**Step 2:** Step 1 is applied to the  $2^b - 1$  symbol Markov source,  $\{S^{2^b-1}, Fr^{2^b-1}, C^{2^b-1}\}$ . So

that the  $2^b-1$  symbol problem is reduced to  $2^{b-2}$  symbol problem.

**Step 3:** Step 2 is repeated and after  $2^b-1$  steps, we obtain  $\{S^2, Fr^2, C^2\}$  a data source of a symbol set  $S^2$ , that has only two symbols in it.

**Step 4:** To the symbol set  $S^2$ , which has two elements, we assign a code of 0 to one symbol and 1 to the other symbol, i.e.  $C^2 = \{c_1^2 = 0, c_2^2 = 1\}$ .

**Step 5:** We build the codes  $C^3 = \{c_1^3, c_2^3, c_3^3\}$  using,  $c_{\sigma^3(1)}^3 = c_1^2$ ,  $c_{\sigma^3(2)}^3 = c_2^2 0$ ,  $c_{\sigma^3(3)}^3 = c_2^2 1$ .

Here,  $c_2^2 0$  and  $c_2^2 1$  denote the string  $c_2^2$  concatenated with 0 and 1 respectively.

Finally, codes  $C = \{c_1, c_2, \dots, c_{2^{b-1}}\}$  are obtained using the relations  $c_{\sigma^n(i)} = c_i^{2^{b-1}}$

for  $i = 1, 2, \dots, 2^{b-2}$ ,  $c_{\sigma^n(2^{b-1})} = c_{2^{b-1}}^{2^{b-1}} 0$  and  $c_{\sigma^n(2^b)} = c_{2^{b-1}}^{2^{b-1}} 1$ .

Either the code book or the frequency set  $Fr$  is stored along with the coded data. In case the code book is not stored, it can be constructed by using  $Fr$ .

### Decoding:

**Step 1:** We read one bit of the coded data and store it as a string  $b_1$ .

**Step 2:** String  $b_1$  is matched with the code book, if a matching occurs  $b_1$  is decoded, else step 3 is executed.

**Step 3:** Another bit is read and concatenated with the string  $b_1$ . Step 2 is repeated until all the bits in the coded data is exhausted.

### Length of Huffman codes:

Let  $l_H$  be the average code length in a Huffman coding for a source  $S$ . Let,  $E(S)$  be the entropy of the source  $S$ , given by equation (1.2.1). Gallager [20] has shown that the average length of Huffman code satisfies the following inequality,

$$E(S) \leq l_H \leq E(S) + p_1 + \log_2 \left[ \frac{2(\log_2 e)}{e} \right],$$

where  $p_1$  is the probability of most likely symbol in the data sequence. There are several modifications to the Huffman coding scheme available in the literature (Kunth [43], Sayood [65]). A combination of pixels considered to constitute one symbol is one such modification to the original Huffman scheme. The average bit rate using modified Huffman coding, approach the true entropy of the image, if combinations of pixels are taken to constitute single symbol. In Huffman coding, we require integral number of bits to represent each symbol. Even if the histogram of an image is highly peaked, this coding would require at least one bit per pixel. Apart from lossless image compression, Huffman coding has applications in text and audio compression.

### 1.2.2 Arithmetic Coding:

Arithmetic coding is another popular method for data compression, and it does not suffer from the limitation that each code word has to be one bit long. There have been many developments in the practical arithmetic coding strategies by Langdon and Rissanen [44]. Here we present the basic algorithm for arithmetic coding.

Let  $S = \{1, 2, \dots, 2^b\}$  be the source set of  $2^b$  symbols, typically representing an image of depth  $b$ . Let

$$f_c : \{1, 2, \dots, 2^b, 2^{b+1}\} \rightarrow [0, 1]$$

be the cumulative distribution function (cdf) of the probabilities  $P = \{p_1, \dots, p_{2^b}\}$  of occurrences of the  $2^b$  symbols in a digital image.

$$f_c(i) = \sum_{j \leq i} p_j, \text{ for } i = 1, 2, \dots, 2^b.$$

Let us assume,  $f_c(0) = 0$  and  $f_c(2^b + 1) = 1$ .

Let  $D = \{d_1, \dots, d_m\}$  be a data sequence of length  $m$ . We need to find the code which is a real number here. The real number obtained in this process is stored in sufficient precision so that the data sequence can be uniquely decoded. The following steps are used,

### Coding:

**Step 1:** Set  $x_0 = 0$ ,  $y_0 = 1$ . For  $k = 1, 2, \dots, m+1$ , the values of  $x_k$  and  $y_k$  are calculated using the relation,

$$\begin{aligned} x_k &= x_{k-1} + (y_{k-1} - x_{k-1}) f_c(d_k - 1) \\ y_k &= x_{k-1} + (y_{k-1} - x_{k-1}) f_c(d_k) \end{aligned} \quad (1.2.2)$$

**Step 2:**  $C = \frac{x_m + y_m}{2}$ , where  $C$  is the code for the input data sequence.

### Decoding:

**Step 1:** We set  $x_0 = 0$  and  $y_0 = 1$ . For each  $i$ , we find  $C^* = \frac{C - x_{i-1}}{y_{i-1} - x_{i-1}}$

**Step 2:** We find the value of  $d_k$  for which  $C^* \in [f_i(d_k-1), f_i(d_k))$ . The values of  $x_k$  and  $y_k$  are updated using the relation (1.2.2).

**Step 3:** Step 2 is repeated until the entire data is decoded.

### Length of Arithmetic codes:

The average length of an arithmetic code  $l_A$  of a data sequence of length  $m$  is given by,

$$E(S_m) \leq l_A \leq E(S_m) + \frac{2}{m},$$

where  $E(S_m)$  is the entropy of sequence of  $m$  symbols. By increasing the length of the sequence, a rate as close to the entropy as we want can be guaranteed. Arithmetic coding method is used in Joint Bi-level Experts Group (JBIG) standard (Langdon [37], Sayood [65], Hunter and Robinson [31]) for encoding binary (two color) images.



### 1.2.3 Dictionary Based Compression:

It is one of the approaches to encode data sources in which certain patterns are frequently occurring. The encoding method is to keep a dictionary of frequently occurring patterns. When these patterns appear in the source data they are encoded with a reference to the dictionary. The principle of this kind of compression is that it encodes variable length strings of symbols as a single token, which are some reference to the dictionary. The objective is to replace certain phrases with tokens. If the number of bits in the token is less than the number of bits in the phrase, compression will occur. There are two approaches for building the dictionary. Static dictionary, where the dictionary is completely defined when encoding begins. This approach would require sufficient prior knowledge about the data. On the other hand, Adaptive dictionary start out with a default baseline dictionary. As the coding proceeds, the algorithm adds new phrases to be used later as encoded tokens.

Certain aspects of statistical modeling, related to entropy, character and word frequencies, were the main areas of research in data compression up to 1977. In the years 1977 and 1978, J.Ziv and A.Lempel [77]-[78], initiated works in dictionary based compression. They developed two algorithms LZ77 and LZ78, and these algorithms triggered research in dictionary based compression. In principle, these are two different techniques.

#### **LZ77 approach:**

The input data sequence is scanned by a sliding window. The window consists of two parts, a search buffer that contains the portion of the recently encoded sequence, and a look-ahead buffer that contains next portion of the sequence to be encoded. The sizes of buffers are implementation dependent and they are generally large. The algorithm moves a search pointer back through the search buffer until it encounters a match. The number of symbols passed by the pointer to reach the match is called offset. The algorithm then examines the symbols following the symbol at the pointer location to see if they match consecutive symbols in the look ahead buffer. Match length  $l$ , is the number of consecutive symbols in the search buffer that matches consecutive symbols in the look

ahead buffer. After finding the longest match for the given window size, the algorithm codes it with a triplet  $\{o, l, c\}$  where  $o$  is the offset,  $l$  is the match length and  $c$  the code word corresponding to the symbol in the look ahead buffer that follows the match.

**LZ78 approach:**

In LZ78, instead of using fixed length phrases from a window into the data sequence, it builds phrases up one symbol at a time, adding a new symbol to an existing phrase when a match occurs. It constructs a dictionary while both coding and decoding. The construction at both ends is done in an identical manner. The inputs are coded as a double  $\{i, c\}$  where  $i$  corresponds to the index to the dictionary entry that was the longest match to the input, and  $c$  being the code for the character in the input following the matched portion of the input. In case of no match the double becomes a new entry in the dictionary. Each new entry into the dictionary is one symbol concatenated with an existing dictionary.

The dictionary building is more implementation based and several variations are possible. Notable among the modifications is the one given by Welch [74], and LZ is modified to LZW algorithm.

**Run Length Coding:**

It is a simple technique to compress large runs of identical symbols in a data sequence, (Capton [11]). RLE encodes a run of symbols as a symbol and a count. It is often used for its easy implementation.

The pixel values in most practical images tend to be well spread out over their entire range. The peaks in histogram may not be found in images for statistical compression to succeed.

### 1.3 Lossy Compression:

The major advantage of the above methods is that, after decompression we get back the original image without any distortion. These types of algorithms are in general referred to **lossless compression**. These methods do not give rise to high compression when applied to image data. Images based on scanned photographs do not possess the characteristics to create multiple occurrences of the same phrase. Out of a sequence of twenty-five pixels say, one or two may vary by a single step from the scans before and after and they often reduce the performance of dictionary or statistically based compression. However, these differences are small enough, to the extent that they are either undetectable or meaningless to human eye. We see that the pixel values necessarily have to follow similar pattern for the dictionary methods to work better. The minute variations which are undetectable to the human eye limits the effectiveness of these compression strategies. Certain mechanism which will enhance the performance of the above said methods are needed.

In order to enhance the performance of the statistical or dictionary methods, models based on human visual systems (HVS), (Jayant, et al. [36]) were brought into the scene. In these models, details which are not detectable by human eye are considered as redundant. Consequently methods which introduce certain distortion between the original image and the decompressed image are developed. The algorithms are modeled in such a way that the distortion cannot be perceived by human eye. These algorithms are based on the philosophy that images are meant for visual inspection by humans, and they cannot perceive certain very fine variations in the pixel values, so it is sufficient to store images up to details which are perceivable by human visual system. These kinds of compressions are referred to as **lossy compressions**. However, the term 'loss' in the lossy compression depends on the application. Medical images help radiologist, in diagnosis, it often imposes stringent conditions on the amount of permitted distortion for the sake of compression. Consequently lossy compression schemes for medical images would be required to perform in a 'nearly lossless' manner. The general principle of this type of compression is predictive or transform coding followed by quantization. The

quantized transform coefficients are retained, from which the decompressed image could be reconstructed at the time of need.

### 1.3.1 Predictive Coding:

Methods which predict the current pixel (the one to be encoded) reasonably well by using the information about the previous pixel values fall in this category. It is based on the statistical dependency of one sample element and the next. Let  $\{x_n\}$  represents a data sequence in which the first  $k-1$  elements are encoded and we are encoding the  $k$  th element. Let  $x_k^*$  be a quantity predicted for the data point  $x_k$  based on the previous  $k-1$  elements, considering  $x_k^*$  as an estimate of  $x_k$ , the prediction error  $e_k$  is defined as,

$$e_k = x_k - x_k^*.$$

It is sufficient to quantize  $e_k$  instead of  $x_k$  the actual data point. If  $e_k^*$  is the quantized value of  $e_k$ , then an approximation the original  $k$ -th data point  $\tilde{x}_k$  could be obtained using,

$$\tilde{x}_k = x_k^* + e_k^*.$$

Common data compression that uses predictive quantization is called Differential Pulse Code Modulation (DPCM) (see, Jain, [35]). The error or loss in the process is given by,

$$x_k - \tilde{x}_k = e_k - e_k^*.$$

### 1.3.2 Transform Coding:

Image data is transformed into a suitable transform domain by using an invertible transformation. Let the digital image be, (cf: 1.1.1)

$$F = \sum_{i,j} P_{ij}(f) X_{ij}(x, y).$$

$P_{ij}(f)$  are stored as pixel values of the image  $f$ . A representation of  $f$  in the transform domain is obtained as  $\tilde{f}$ , given by

$$\tilde{f} = \sum_{i,j} \tilde{C}_{ij}(f) \phi_{ij}, \quad (1.3.1)$$

where  $\tilde{C}_{ij}(f)$  are the transform coefficients of  $f$  and  $\{\phi_{ij}\}$  is an orthonormal basis of the transformation. In transform coding, a discrete transform based on  $\{\phi_{ij}\}$  is applied over

the pixel data to get the transform coefficients. Instead of storing the transform coefficients  $\tilde{C}_{ij}(f)$ , quantized coefficients  $C_{ij}^*(f)$  are stored.  $\tilde{C}_{ij}(f)$ 's are obtained by quantizing  $\tilde{C}_{ij}(f)$ 's. Some of the useful candidates for the transform are listed here.

### Karhunen-Loeve Transform (KLT):

It is the optimal transform coder (see Donoho, et al. [19]), in the sense that it minimizes the mean square error of the decompressed data for a given number of total bits. Consider a random vector,

$X = [X_1, \dots, X_n]^T$  with zero mean. The vector  $X$  is linearly transformed by a matrix  $A$  to produce a vector  $Y = [Y_1, \dots, Y_n]^T$ . The auto covariance matrices  $R_x$  and  $R_y$  are defined as,

$$R_x = E(XX^T) \text{ and } R_y = E(YY^T).$$

The transformation matrix  $A$  is chosen such that the components of the resulting vector  $Y$  are mutually uncorrelated. After obtaining  $Y$ , it is subjected to component wise quantization to obtain  $\tilde{Y}$  an approximation to  $Y$ . The vector  $\tilde{Y}$ , is linearly transformed by a matrix  $B$  to get  $\tilde{X}$ . Matrices  $A$  and  $B$  are obtained so that the quantity,

$$\frac{1}{N} E \left[ \sum_{i=1}^N (X(i) - \tilde{X}(i))^2 \right] \text{ is minimized.}$$

The matrix  $R_x$  is symmetric and positive semi definite, there exists a full set of eigen vectors with non negative eigen values. The KL transform matrix  $A$  for the data sequence  $X$  is constructed by keeping the orthonormalised eigen vectors of the auto covariance matrix  $R_x$ . The rows of  $R_x$  are ordered in terms of decreasing eigen values, that is  $R_x A = A \Lambda$  where  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ . Transforming  $X$  with  $A$  will diagonalise  $R_y$ .

$$R_y = E[A^* X X^T A] = A^* E(XX^T) A = A^* R_x A = \Lambda.$$

KLT satisfies the best linear approximation property in the  $l_2$  sense. When the number of transform coefficients to be retained are fixed, then KLT is the optimal transform. However the algorithm complexity involved in KLT, restricts its practical usage. It requires  $N^2$  operations per vector of length  $N$ . If we view the image as a  $N \times N$  matrix, comprising of  $N$  row vectors or  $N$  column vectors, then KLT would require  $N^3$  operations on these  $N$  vectors. Also KLT is dependent on the data sequence to be compressed. Due

to these reasons, fast fixed transforms, which work at  $N^2 \log N$  complexity, are practically used for compression. Discrete Fourier transforms, Discrete Cosine transform, Discrete Sine transform, Hadamard transform etc., are being substituted for KLT, because of their implementation efficiency. The following transforms are commonly used (Jain, [34]).

### Discrete Fourier Transform:

The DFT forward and inverse for an image is given by,

$$\begin{aligned} \text{DFT}(m, n) &= \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \text{Pix}(x, y) e^{-\frac{i(mx+ny)2\pi}{N}} \\ \text{Pix}(m, n) &= \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \text{DFT}(x, y) e^{\frac{i(mx+ny)2\pi}{N}} \end{aligned} \quad (1.3.2)$$

Although the Fourier transform produces complex transform coefficients in general, for real input sequences, it is sufficient to store half the data and the rest can be generated by using the symmetry properties.

### Discrete Cosine Transform:

$$\text{DCT}(i, j) = \sqrt{\frac{2}{N}} C(i) C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \text{Pix}(x, y) \cos\left[\frac{(2x+1)\pi i}{2N}\right] \cos\left[\frac{(2y+1)\pi j}{2N}\right] \quad (1.3.3)$$

$$\text{Pix}(i, j) = \sqrt{\frac{2}{N}} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} C(x) C(y) \text{DCT}(x, y) \cos\left[\frac{(2x+1)\pi i}{2N}\right] \cos\left[\frac{(2y+1)\pi j}{2N}\right] \quad (1.3.4)$$

$$\text{where } C(x) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}$$

DCT performed on an  $N \times N$  square matrix of pixel values yields an  $N \times N$  square matrix of frequency coefficients. DCT can be actually computed like FFT and can be done in  $O(N^2 \log N)$  complexity.

### Discrete Sine Transform:

The sine transform and inverse sine transform pair for each row of the image is given by,

$$\begin{aligned}
DST(j, k) &= \sqrt{\frac{2}{N+1}} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} \text{pix}(m, n) \sin \frac{\pi(j+1)(m+1)}{N+1} \sin \frac{\pi(k+1)(n+1)}{N+1} \\
&\quad 0 \leq k \leq N-1, 0 \leq j \leq N-1 \\
\text{Pix}(m, n) &= \sqrt{\frac{2}{N+1}} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} DST(j, k) \sin \frac{\pi(j+1)(m+1)}{N+1} \sin \frac{\pi(k+1)(n+1)}{N+1} \\
&\quad 0 \leq m \leq N-1, 0 \leq n \leq N-1
\end{aligned} \tag{1.3.5}$$

### Hadamard Transform:

$$\begin{aligned}
HT(j, k) &= \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} \text{pix}(m, n) (-1)^{b(k, m) + b(k, n)} \quad 0 \leq j \leq N-1, 0 \leq k \leq N-1 \\
\text{Pix}(m, n) &= \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} HT(j, k) (-1)^{b(k, m) + b(k, n)} \quad 0 \leq m \leq N-1, 0 \leq n \leq N-1
\end{aligned} \tag{1.3.6}$$

where,

$$b(k, r) = \sum_{i=0}^{n-1} k_i r_i \quad \text{with } k_i, r_i = 0, 1 \text{ and } n = \log_2 N. \quad \{k_i\} \text{ and } \{r_i\} \text{ are the binary}$$

representations of  $k$  and  $r$ , respectively, that is,

$$k = k_0 + 2k_1 + \dots + 2^{n-1}k_{n-1} \quad \text{and} \quad r = r_0 + 2r_1 + \dots + 2^{n-1}r_{n-1}.$$

The above mentioned transforms falls in the category of fast transforms. These transforms can be calculated via FFT, and their computational complexity is  $O(N^2 \log N)$  for  $N \times N$  images. These transforms posses good energy compaction property for highly correlated images. DCT is a most widely used transform for image compression.

### 1.3.3 Quantization:

After the application of transform, the coefficients are subjected to quantization which actually causes the compression. Quantization, (Gray, R.M., etal., [23]) or round off is a technique originated from analog to digital conversion of signals. It amounts to partitioning the range into small sub intervals (subsets) and choosing a representative value for each sub interval. Any data point that occurs from the subinterval is quantized by the representative value. More precisely, a quatizer is a set of intervals or cells,  $S = \{$

$s_i : i \in Z^+$  }. Together with a set of representative values  $C = \{y_i : i \in Z^+\}$  so that the overall quantizer  $q$  is defined by

$$q(x) = \sum_i y_i X_{s_i}(x), \quad (1.3.7)$$

where  $X_{s_i}(x)$  is the indicator function for the cell  $s_i$ .  $S$  is the partition of the original range. Uniform quantizer is one in which the levels are equi spaced and the representative value is the mid point.

The goodness of reproduction measures the quality of a quantizer. If  $x$  is the input signal and  $\tilde{x}$  is the output signal after quantization, the error due to quantization in  $l_2$  sense is given as

$$q-l_2 = E[(x - \tilde{x})^2].$$

The quantizer that minimizes the  $l_2$  error for a given number of quantization levels is the Lloyd-Max quantizer (Jain, [34]). The concept of scalar quantization, where the quantization is done component wise has been extended to vector quantization (Gersho and Gray [23]), it is just like scalar quantization except that all components of a vector, say  $k$  successive source samples are quantized simultaneously. They are characterized by a  $k$ -dimensional partition, a  $k$ -dimensional code book (containing  $k$ -dimensional points, reconstruction code words or code vectors) and an assignment of binary code words to the cells of partition.

## 1.4 Standard Compression Methods:

In the present section, we give a brief introduction to the standard compression methods. These methods are compared with our strategies in the thesis.

### 1.4.1 JPEG Standard:

Constructive Committee on International Telephone and Telegraph (CCITT) and International Standards Organization (ISO), in 1980's constituted a committee - Joint Photographic Experts Group (JPEG), with an idea to develop a standard for image compression. The committee came up with JPEG standard, which has been recognized



and used as a major portable graphical file format (Wallace, [73]). The JPEG committee designed a standard for compressing either full color or gray scale still images of natural, real world scenes. JPEG is designed to exploit the known limitations of the human eye, it falls in the category of lossy image compression. However, the standard allows the user to choose the amount of permitted loss. The degree of loss could be adjusted using a compression parameter called quality factor. The JPEG specification is broadly divided into lossless and lossy encoding modes.

The lossless compression uses the predictive or adaptive model described earlier, with a Huffman code output. It does not allow any distortion in the decompressed image. The compression efficiency in terms of the size of the compressed image is not satisfactory for many applications. On the other hand, the JPEG lossy compression algorithm has good compression efficiency. JPEG has three major parts - transform coding using Discrete Cosine Transform (DCT) (1.3.3)-(1.3.4), quantization and loss less compression of quantized DCT coefficients.

DCT applied to the pixel data prepares the data for quantization. The amount of quantization will depend on the required quality of the output. Coefficients which are sufficiently close to zero are truncated to zero. The quantization step makes the DCT coefficients matrix sparse. Run length coding (RLE) is used to absorb the zero runs of the coefficient matrix. In order to increase the length of runs, the coefficient matrix is first rearranged to form a zig-zag sequence. Instead, of compressing the coefficients in a row major fashion, the zig-zag scanning moves along diagonal paths which progressively moves from low frequency to high frequency terms. Since the probability of a high frequency term rounding off to zero in the quantization step is more, the coefficients are arranged in a frequency order so that the indexing will be minimal and easier when storing the quantized DCT coefficients.

Finally an entropy encoder (either Huffman coding or Arithmetic coding) depending on the choice of the implementation is used. To enhance the speed of operations block

coding is done. In block coding, the whole image is partitioned into  $8 \times 8$  blocks and the DCT, quantization and coding are applied for each block independently.

Although JPEG works well for images of natural and realistic scenes. It does not work well for line drawings, cartoons and sharp edges in images tend to come out blurred. Human visual system is more sensitive to brightness variations than to hue variations. JPEG can compress hue data more heavily than brightness (gray-scale) data. Further JPEG accumulates losses, in other words recompressing a decompressed image loses more information.

Another generation of image coders mostly based on wavelet decompositions, other fast transform based decompositions, different quantization techniques and entropy coding are being considered for the next standard JPEG-2000 (see Donoho, et al. [19]).

#### **1.4.2 Graphics Interchange Format (GIF):**

It is another popular commercial compression standard (Sayood, [65]). It uses LZW compression strategy (Welch, [74]) and certain variations of that. It is suitable for images that contain large regions of constant value. GIF does significantly well on images, which have few distinct levels. GIF is lossless for images having 256 distinct states (gray levels), which is the case of most practical interest in medical imaging.

#### **1.4.3 Fractal Compression:**

Objects which have self similar structures are called fractals. The main idea behind the fractal compression is to use the similarities of certain regions of object with itself. A mapping  $f$  from a set to itself is said to be contractive if  $d(f(x), f(y)) \leq d(x, y)$  where  $d$  is distance measure. It is well known that a contractive mapping has a fixed point (Rudin, [64]). Suppose that for a given image  $I$ , there exists a function  $f$  such that  $f(I) = I$ . Then  $f$  can be used for representation of the image  $I$ . Fractal compression amounts to finding a function  $f$  for the given image  $I$ , so that  $I$  is a fixed point of  $f$ . Arnaud Jacquin [33] in 1989 came up with an idea of partitioning the image  $I$  into sub images  $I_k$ , and for each  $I_k$  a suitable function  $f_k$  is obtained by using a method called Iterated Function System (IFS)

(Y. Fischer [76]). These types of compression strategies are of recent origin and a lot of research needs to be done based on these ideas.

#### 1.4.4 Wavelet based Compression:

In 1992,. R.A.DeVore, B.Jawerth and B.J.Lucier [17] have proposed a compression algorithm based on wavelet decomposition. They associate to each set of pixel values a function  $f$  given by,

$$f = \sum_{k \geq 0, j=(j_1, j_2)} c_{j,k} \phi_{j,k}, \quad (1.4.1)$$

where  $c_{j,k}$ 's are the wavelet coefficients of  $f$ .  $\phi_{j,k}$ 's are generated from a single function  $\phi$  by their dyadic dilates and translates given by,

$$\phi_{j,k}(x) = \phi(2^k x - j). \quad (1.4.2)$$

After obtaining a wavelet-based decomposition of the form (1.4.1), their compression algorithm quantizes the transform coefficients to obtain quantized coefficients  $\tilde{c}_{j,k}$  and the compressed function takes the form,

$$\tilde{f} = \sum_{k \geq 0, j=(j_1, j_2)} \tilde{c}_{j,k} \phi_{j,k}. \quad (1.4.3)$$

Their theory relates the smoothness of images in certain smoothness classes called Besov Spaces [16], which are defined as follows:

#### Definition (1.4.4) (Besov Spaces):

Fix  $\alpha > 0$  and an integer  $k > 0$ . Define the  $k$ -th forward difference with a parameter  $h \in \mathbb{R}^+$  by

$$\Delta_h^0 f(x) := f(x)$$

and for  $j = 1, \dots, k$   $\Delta_h^j f(x) := \Delta_h^{j-1} f(x+h) - \Delta_h^{j-1} f(x)$ .

For  $1 \leq p < \infty$ , the  $k$ -th modulus of smoothness in  $L_p(I)$  is defined as

$$\omega_k(f, t)_p = \sup_{|h| \leq t} \left( \int_{I_n} |\Delta_h^k f(x)|^p dx \right)^{\frac{1}{p}}$$

and for  $p = \infty$ ,

$$\omega_k(f, t) := \sup_{|h| \leq t} (|\Delta_h^k f(x)|).$$

where  $I_{rh} = \{x \in I : x+rh \in I\}$ .  $\Delta_h^k f(x)$  is defined iff  $x \in I_{rh}$ . The Besov space,  $B_q^{\alpha,k}(L_p(I))$  norm is defined as

$$\|f\|_{B_q^{\alpha,k}(L_p(I))} = \|f\|_p + \left( \int_0^\infty \left[ t^{-\alpha} \omega_k(f, t)_p \right]^q \frac{dt}{t} \right)^{\frac{1}{q}}$$

for  $1 \leq p \leq \infty$ ,  $0 < q < \infty$  and for  $q = \infty$ ,

$$\|f\|_{B_q^{\alpha,k}(L_p(I))} := \|f\|_p + \sup_{t>0} [t^{-\alpha} \omega_k(f, t)_p]$$

$f \in B_q^{\alpha,k}(L_p(I))$  if  $\|f\|_{B_q^{\alpha,k}(L_p(I))} < \infty$

The error function in  $L_p(I)$  is given by,

$$a_N(f)_p = \inf_{\tilde{f} \text{ has } \leq N \text{ coefficients}} \|f - \tilde{f}\|_{L_p(I)}. \quad (1.4.5)$$

For certain choices of  $\phi$ , DeVore, Jawerth and Popov [18] have shown that,

$$a_N(f)_p = O(N^{-\alpha/2}) \quad \Leftrightarrow \quad f \in B_q^{\alpha,k}(L_q(I)) \quad (1.4.6)$$

where  $\frac{\alpha}{2} + \frac{1}{p} = \frac{1}{q}$ .

They have proposed a compression algorithm based on the wavelet decomposition of  $f$ . The quantization strategy to obtain  $\tilde{c}_{j,k}$ 's is that they satisfy,

$$\|(c_{j,k} - \tilde{c}_{j,k})\phi_{j,k}\|_{L_p(I)} \leq N^{-1/q}. \quad (1.4.7)$$

Here  $N$ , the quantization parameter, is a chosen positive integer, and it determines the amount of compression. The number  $N$ , of nonzero coefficients  $\tilde{c}_{j,k}$ 's is shown to satisfy,

$$N \leq C_1 N \|f\|_{B_q^{\alpha,k}(L_q(I))}^q. \quad (1.4.8)$$

## 1.5 MRI image compression:

Images arising from various sources form different classes. For example images of natural scenes, machine parts, buildings, person's etc., fall in different categories of images and the corresponding redundancy models are different. In the present thesis, we deal with the images arising from MRI. Apart from comparing a number of image compression methods, we propose newer methods for compression, which are well suited for MRI images. We study the reliability of the proposed compression methods. The performances of these methods are evaluated theoretically and experimentally. The goal of compression is to represent MRI images with the smallest possible number of bits, thereby minimizing storage requirements for archiving and speeding transmission for the sake of teleradiology (Gitlin, [22]). A compression system typically consists of one or more of the following units,

- (i) Approximation of the underlying intensity distribution function, by a suitable approximation process.
- (ii) Quantization of the analog or high rate digital pixels, into a relatively small number of bits. This conversion is lossy.
- (iii) Lossless compression achieved by invertible methods such as Huffman coding, Arithmetic coding, LZ coding (cf: section 1.2). One of the lossless compression methods is applied over the quantized data for obtaining net compression.

The above units are studied for MRI images, in which not much loss of information can be tolerated in compression/decompression for most medical applications (Cosman, et al., [13]). Most of the existing methods produce compressed images, which are to be used for visual inspection, and they use the limitations of Human Visual System (HVS). However, for MRI images, there are certain other quantities that can be computed for the whole image or for some sub regions, depending on the need. These quantitative informations along with the images are required in diagnostic studies, especially in cancer tissue characterization and to study the effectiveness of the treatment. Images produced using our compression strategy can be used for one or more of the following applications:

- (c1) Estimation of tissue parameters,  $\rho$ ,  $T_1$  and  $T_2$  (studied more precisely in the next section) which are used for certain diagnostic studies.
- (c2) Volume or area estimation of geometric structure typically cancerous portions seen in MRI images.
- (c3) Visual inspection by experts, for diagnostic purposes.

We note that the computations involved in (i) are highly sensitive to the input data. They require large set of input data and hence compression will be useful. Certain MRI preliminaries are presented in the following sub section.

### 1.5.1 Magnetic Resonance Imaging - Basics:

Magnetic Resonance Imaging (MRI) is a technique for providing a picture of the interior of a physical object. Magnetic resonance (MR) is based on the interaction between a system of atomic nuclei and an external magnetic field. In MRI, the system is perturbed from the equilibrium and the system relaxation to the equilibrium is observed. An induced electric signal is measured during this relaxation process. This signal depends on four quantities  $\rho$ ,  $T_1$ ,  $T_2$  and  $P$  (a parameter set), which measure the nuclear spin density, the interactions of the nuclei with their surrounding molecular environment, those between close nuclei, and certain pulse sequence parameters respectively. Based on a suitable set of measured electric signals, a function of  $\rho$ ,  $T_1$ ,  $T_2$  and  $P$  is computed, which is called MR image. Basic references describing MRI are available in (Healy, et al., [26], Hinshaw, et al., [29], Kumar, A., et al., [42], Mansfield, et al., [51], [52], Sebastiani et al., [66], Stark et al., [69]. Here we briefly present certain MRI preliminaries, which are used in the present thesis.

The hydrogen nuclei, often referred to as spins, can be imagined as tiny bar magnets. In the absence of an external magnetic field, spins assume random orientations, and the net magnetization will be zero. When exposed to a static magnetic field  $H_0$ , the randomly oriented nuclei lineup with the magnetic field. The spin vectors experience a torque or couple when subjected to the magnetic field, analogous to a spinning top in the earth's

gravitational field. As a result they precess around the axis of the magnetic field at a rate given by the Larmour relationship (see Stark et al., [69]),

$$f = \frac{\omega}{2\pi} = \frac{\gamma H_0}{2\pi}, \quad (1.5.1)$$

where  $f$  is the resonance frequency in Hertz,  $\omega$  is the angular frequency in radians per second,  $\gamma$  is the gyromagnetic ratio, which is the characteristic of every isotope and  $H_0$  is the static magnetic field. Let the vector sum of the individual spins be approximated by the net magnetization vector of the sample,  $M$ . The net magnetization  $M$ , is the actual observable property of the sample.

In a typical MR experiment, the direction in which the net magnetization vector  $M$  is pointing is disturbed. Then it relaxes back to its equilibrium position. The magnetic field  $H_0 = H_0 k$  is applied in the  $z$ -direction,  $k$  is the unit vector in the  $z$  direction. Before excitation,  $M$  is directed along the  $z$ -axis and therefore has no component in the  $xy$  plane. The excitation process is tipping  $M$  away from the equilibrium position along  $z$ . This is achieved by temporarily applying a second, much weaker magnetic field  $H_1$ , perpendicular to  $H_0$ . Also  $H_1$  rotates about  $H_0$  at the Larmor frequency.  $H_0$  is normally in the range 0.1 - 5.0 Tesla,  $\omega_0$  is in MHz range, which is the radio frequency band of the electromagnetic spectrum. The brief application of  $H_1$  rotating at  $\omega_0$  is called RF pulse.

To understand the effect of the RF field  $H_1$  on the net magnetization vector  $M$ , it is convenient to introduce the concept of rotating frame of reference (see, Hinshaw and Lent [29]). This is simply a frame of reference which rotates around the static magnetic field  $H_0$  at the Larmor frequency  $\omega_0$ . In the rotating frame of reference,  $M$  does not appear to precess since the observer also rotates. To achieve a  $90^\circ$  rotation, i.e.  $90^\circ$  excitation pulse, the RF amplitude  $H_1$  and duration  $\tau$  can be adjusted to give  $\gamma H_1 \tau = \pi/2$ . At the end of such excitation pulse,  $M$  would be in the transverse plane and would appear stationary in the rotating frame of reference. In the laboratory frame of reference the rotation of  $M$  away from  $z$  follows a spiral trajectory over the upper half of a sphere. At the end of RF pulse,  $M$  will precess freely about  $z$  at its Larmor frequency. The signal generated by the magnetization after a RF pulse is called a **Free Induction Decay (FID)**.

The free precession around  $z$  continues as the system returns to its equilibrium position. The behavior of  $M$  during this portion of the MR experiment is characterized by two processes, spin-lattice and spin-spin relaxation, which have exponential time constants  $T_1$  and  $T_2$  and describe the decay and recovery of the longitudinal ( $z$ ) and transverse ( $x$  and  $y$ ) components of  $M$ , respectively. The transverse component of  $M$  precessing in an ever-decreasing radius circle in the transverse plane (spin-spin relaxation) while  $z$  component of  $M$  grows with time (spin-lattice relaxation). Because of the nature of the underlying process  $T_2 \leq T_1$  (Peters and Williams [58]). In biological tissues at typical imaging field strengths (0.1 – 2.0 Tesla),  $T_1$  values are of the order of 0.1-50s and  $T_2$  is 50-1000 ms.

It is an experimental fact that the motion of  $M(t)$  for  $t \geq t_0$  is a damped precession around  $z$ -axis, with the Larmor frequency, and  $M(t) \rightarrow M_0 k$  as  $t \rightarrow \infty$ , due to spin-lattice and spin-spin interactions. In 1946, F.Bloch [6] formulated the differential equations describing the motion of the components of the magnetization  $M$ . For the class of experiments in which the magnetic field is a small perturbation of a static and homogeneous field, i.e  $H(x,t) = H(t) = H_0 k + H_1(t)$ , with  $H_0 \gg \max_t |H_1(t)|$ , and in the case of  $M(x,t) = M(t)$  the following equations are basic to MR experiments and are called Bloch equations.

$$\frac{d}{dt} M(x, t) = \gamma M(x, t) \otimes H(x, t) - \frac{M_x i + M_y j}{T_2} - \frac{(M_z - M_0)k}{T_1}. \quad (1.5.2)$$

The notation,  $\otimes$  in (1.5.2) refers to cross product. Component wise,

$$\begin{aligned} \frac{d}{dt} M_x(t) &= \gamma [M_y(t)H_z(t) - M_z(t)H_y(t)] - \frac{M_x(t)}{T_2} \\ \frac{d}{dt} M_y(t) &= \gamma [M_z(t)H_x(t) - M_x(t)H_z(t)] - \frac{M_y(t)}{T_2} \\ \frac{d}{dt} M_z(t) &= \gamma [M_x(t)H_y(t) - M_y(t)H_x(t)] - \frac{M_z(t) - M_0}{T_1} \end{aligned} \quad (1.5.3)$$

Bloch equation gives a description of the time dependence of the nuclear magnetization  $M(t)$  in the presence of an applied magnetic field  $H(t)$ . The local inhomogeneities in the magnetic field  $H_0 k$  will also cause  $M_{xy}(t)$  to decrease to zero. Denoting  $T_2$  by  $T_2^*$  which also takes into account the magnetic field inhomogeneities due to which the fall to zero of



$M_{xy}(t)$  will be faster, that is  $T_2^* \leq T_2$  (Sebastiani and Barone [66]). To incorporate this fact in the Bloch equations, let  $W(x,t)$  be the vector with components

$$\left( -\frac{M_x(x,t)}{T_2^*(x)}, -\frac{M_y(x,t)}{T_2^*(x)}, -\frac{M_z(x,t) - M_0(x)}{T_1(x)} \right).$$

We assume that relaxation's due to spin-lattice and spin-spin interactions and magnetic field imperfections can be taken into account by adding to the right-hand side of (1.5.2) the term  $W(x,t)$ . Then the Bloch equations become,

$$\frac{d}{dt} M(x,t) = \gamma M(x,t) \otimes H(x,t) + W(x,t) \quad (1.5.4)$$

where

$H(x,t) = H_0 + H_1(x,t)$  with  $H_0 \gg \max_{x,t} |H_1(x,t)|$ . These Bloch equations provide us with a macroscopic model for the net nuclear magnetization  $M$ .

A technique known as **spin echoing** (see Partain et al., [56]), may be used to eliminate the dephasing effects, caused due to spatial inhomogeneities. Here a 180-degree pulse applied for sometime  $\tau$  after the 90-degree pulse and it can be shown to reestablish the phase coherence  $\tau$  msec later (Stark et al., [69]). Immediately following the 90 degree RF pulse, at  $\tau = 0$ , magnetization is coherent (in phase) in the transverse  $xy$  plane. At time  $t = \tau$  following the 90 degree pulse, the partly dephased transverse magnetization is turned into mirror image positions by 180 degree RF pulse. The fast moving spins catch up with the slow moving spins and refocusing takes place at time  $t = 2\tau$ . The time  $2\tau$  is referred to as the echo time TE. After several repetitions of 90-180-spin echo, the magnetization strength reaches steady state.  $M$  is detected at some point during the relaxation process of each repetition. The repetition time is usually referred to as TR.

In 1972 Lauterbur [45] first showed that by superimposing linear field gradients on the main field, projections of an object could be generated from which an image can be reconstructed. Application of gradient alters the magnetic field in a selected direction.

Let  $G_x = \frac{\partial H}{\partial x}$  be the gradient applied in  $x$  direction, the resonance frequency becomes dependent on the location of the volume element of interest with respect to the  $x$

direction. Both the gradient field  $G$  and the RF field  $H_1$  are under the control of the experimenter. The gradient field is controlled by three input signals, called  $G_x(t)$ ,  $G_y(t)$  and  $G_z(t)$ , while the RF field is controlled by  $H_1(t)$ . These signals are grouped into a four component vector  $P(t)$ - the pulse sequence (Partain et al., [56]). The coils which generate these gradients in the machine are known as the gradient coils. The ideal x-gradient coil causes the z-component of the magnetic field to vary linearly with  $x$  as follows,

$$H(t, x) = H_0 k + G_x(t) x k.$$

The resonance frequency  $\omega_0$  is a function of position,

$$\omega_0(r, x) = \gamma(H_0 + r G_r(t)).$$

The real x-gradient coil produces a field which has components in the  $x$  and  $y$  directions, but the uniform field in the  $z$ -direction is so strong that these components may be neglected. We can assume that  $G_x(t)x$  is the x-gradient field produced by  $G_x(t)$  the x-gradient. This is true for  $y$  and  $z$  gradients also. When all the three gradient coils are turned on, their superimposed fields yield

$$H(t, x) = H_0 k + G_x(t) x k + G_y(t) y k + G_z(t) z k.$$

The three gradients may be formally grouped into a gradient vector  $G(t)$  with components  $G_x(t)$ ,  $G_y(t)$  and  $G_z(t)$ , and we can write

$$H(t, x) = (H_0 + G(t).x) k.$$

The RF field  $H_1(t)$  in the rotating frame of reference is given by,

$$H_1(t) = H_1 [\cos \omega t i + \sin \omega t j],$$

where  $H_1$  is the magnitude of the rotating field.

Let  $R(x) = \int_0^t G(x, \tau) d\tau$ , where  $G(x, \tau)$  is the gradient applied in the  $x$  direction.

The net magnetic field in rotating frame of reference is

$$H(x, t) = (H_0 + R) k + H_1(t) = (H_0 + R) k + H_1 \cos \omega_r t i + H_1 \sin \omega_r t j,$$

where  $\omega_r$  is the angular frequency of the rotating frame of reference. Substituting the net  $H$  which includes the effect of gradient fields, RF field in the rotating frame of reference and the main field  $H_0$ , in (1.5.4), evaluating the cross product and equating the respective vector components we get

$$\begin{aligned}
\frac{dM_x}{dx} &= \gamma [M_y (H_0 + R) - M_z H_1 \sin \omega_r t] - \frac{M_x}{T_2^*} \\
\frac{dM_y}{dx} &= \gamma [M_z H_1 \cos \omega_r t - M_x (H_0 + R)] - \frac{M_y}{T_2^*} \\
\frac{dM_z}{dx} &= \gamma [M_x H_1 \sin \omega_r t - M_y H_1 \cos \omega_r t] - \frac{M_z - M_0}{T_1}
\end{aligned} \tag{1.5.5}$$

$M(t)$  obtained by solving the above equations under the experimental conditions will give the signal. The signal obtained depends on the applied pulse sequence. Consider Spin Echo (SE) sequence,  $90^\circ - \frac{TE}{2} - 180^\circ - \frac{TE}{2} - \text{measure}$ , where TE is the echo time. The repetition time TR, is the time taken per pulse sequence. Let  $M_L$  and  $M_T$  denote the longitudinal and the transverse components of  $M$ . Initially  $M$  is aligned with  $H_0$ , the static magnetic field, so that  $M_L = M$  and  $M_T = 0$ . Following the  $90^\circ$  pulse,  $M_L = 0$  and  $M_T = M$ . The relaxation  $T_1$  and  $T_2$  continues for a time  $\frac{TE}{2}$ .

The signal intensity  $I_{SE}$  corresponding to the SE sequence is given by (Partain et al., [56]),

$$I_{SE} = kM \left( 1 + e^{-\frac{TR}{T_1}} - 2e^{-\frac{(TR-TE/2)}{T_1}} \right) e^{-\frac{TE}{T_2}} \tag{1.5.6}$$

In the above equation  $k$  is proportionality constant. The amount of signal coming from the volume of the sample is a function of both tissue parameters  $T_1$  and  $T_2$ , and the experimental parameters TR and TE.  $M$  is proportional to the number of affected spins in each pixel, it is often called spin density (or proton density), denoted by  $\rho(x,y)$ . Observed contrast in the image is due to a weighted sum of  $\rho$ ,  $T_1$  and  $T_2$ .

$$\text{Constrast} = W_M \left( \frac{\Delta M}{M} \right) + W_{T_1} \left( \frac{\Delta T_1}{M} \right) + W_{T_2} \left( \frac{\Delta T_2}{M} \right), \text{ where } W \text{ is the weighting factor.}$$

For a spin echo image,

$$\begin{aligned}
W_M &= I_{SE} = kM \left( 1 - e^{-\frac{TR}{T_1}} \right) e^{-\frac{TE}{T_2}} \\
W_{T_1} &= -kM \left( \frac{TR}{T_1} \right) e^{-\frac{TR}{T_1}} e^{-\frac{TE}{T_2}} \\
W_{T_2} &= kM \left( 1 - e^{-\frac{TR}{T_1}} \right) \left( \frac{TE}{T_2} \right) e^{-\frac{TE}{T_2}}
\end{aligned} \tag{1.5.7}$$

$W_M$  is large if TR is long and TE is small.  $W_{T_1}$  is large when  $TR=T_1$  and TE is small. And,  $W_{T_2}$  is large when TR is large and  $TE=T_2$ . By choosing a long TE and a long TR, the contrast in the resulting image will depend mainly on  $T_2$ . Images constructed from signals, measured at a long TE and long TR are usually referred to as  $T_2$  weighted images, while a short TE and short TR results in  $T_1$  weighted images, (see page 34 of this thesis, Image 1.5.5 Test image data set 3).

The signal received is the sum of all individual signals from the distribution. Further, the observed transverse magnetization is directly proportional to the spin density, represented by  $\rho(x,y)$ . MR signals obtained correspond to the whole volume of the object. Images corresponding to certain slices are useful in diagnosis. This amounts to isolate a slice parallel to the xy plane at a height  $z_0$ . The slice selection is achieved by selective excitation (Sebastiani et al., [66]). By the application of selective pulse, the required cross-section can be selectively excited. The received signal corresponds to the selected slice. In MR imaging the main objective is to discriminate the variations of spin density within the selected slice as a function of position in the transverse plane. Frequency encoding is used to determine the variation of  $\rho(x,y)$  along x axis. Turning the x gradient after excitation, during the spin echo does frequency-encoding (Stark et al., [69]). Application of x gradient breaks the selected slice into lines of spins transverse to the x-axis. Each line is comprised of all spins whose x coordinates have a given value. Images can be obtained, from frequency encoded data by using projection reconstruction methods developed by Lauterbuier [45]. However, a most of the modern imaging systems uses a technique called spin warp imaging (see Mansfield et al., [50]). Here phase encoding is used to determine the variation of  $\rho(x,y)$  along y axis. Phase encoding of the

y-axis requires many echoes to be acquired. The y gradient is turned on for some time  $t$  at a strength, which varies from one echo to another. During the phase encoding stage, the lines of spins corresponding to different values of the y coordinate are spinning at different rates, due to the applied y gradient. At the time when the y gradient is turned off, the signal received corresponds to varying frequencies in x direction and varying phases in y-direction of the spins.

Consider 2D distribution of spins, the total signal  $S(t)$ , is the sum of all individual signals from the distribution, as represented by,

$$s(t) = \iint_{xy} I_{SE}(x, y) e^{-\frac{t}{T_2}} e^{-i(x.k_x(t)+y.k_y(t))} dx dy, \quad (1.5.8)$$

where  $I_{SE}$  given by (1.5.6), is valid for each point of the cross-section.

$$\text{In (1.5.8), } k_x(t) = \gamma \int_0^t G_x(\tau) d\tau \text{ and } k_y(t) = \gamma \int_0^t G_y(\tau) d\tau.$$

We note that the MRI raw signal is the Fourier transform of the object evaluated at the Fourier space (k-space) coordinates  $k_x(t)$  and  $k_y(t)$ . Signals are being measured while gradients are applied, the number of phase cycles multiplying the spin distribution is constantly changing. Therefore, at each instant, the measured signal represents a single sample of the 2D Fourier transform on k-space. By appropriately designing the pulse sequence (see Partain et al., [56]), these samples can be made to cover all of k-space.

## Raw data and Image

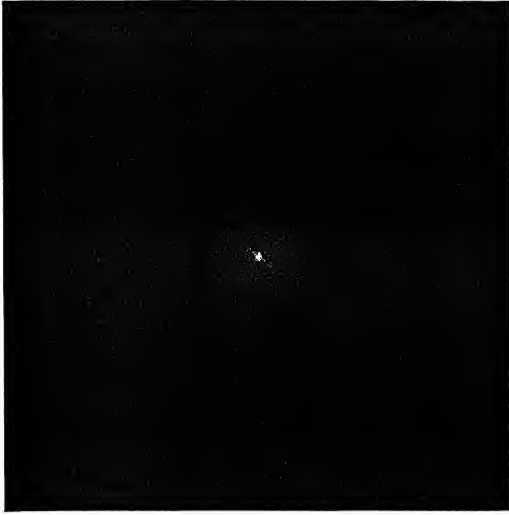


Image 1.5.1

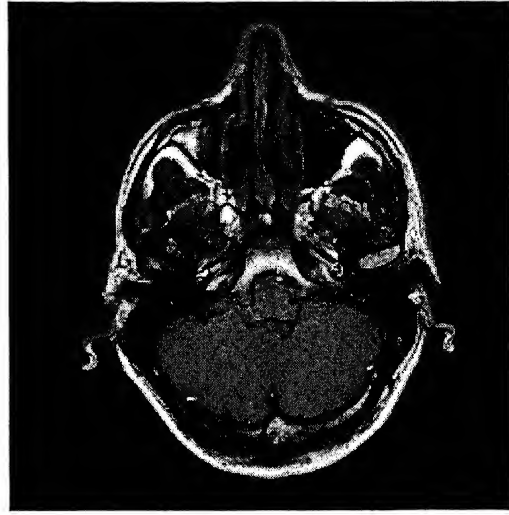


Image 1.5.2

Image 1.5.1 is the magnitude of the raw data and 1.5.2 is the magnitude image obtained from the raw data. We see that in the raw data maximum information is around the center, i.e. low frequency components have higher magnitude than the high frequency components. Based on the frequency components, a variable length coding (VLC) could be devised so that the low frequency components are stored at a higher precision than the high frequency components, resulting in compression.

### 1.6 Experimental Setup:

All the experiments presented in this thesis are done on the actual raw data and images obtained from MRI machine, at Sanjay Gandhi Post Graduate Institute of Medical Sciences (S.G.P.G.I.M.S), Lucknow, India. For most clinical applications, 256 x 256 images are used and our experiments are performed on 256 x 256 image data. We use three different image data sets for three different applications [cf: 1.8 (c1),(c2),(c3)]. We use 2 byte images (depth, 16 bits per pixel) for images to be used for quantitative

estimations and 1 byte image (depth, 8 bits per pixel) for images to be used for visual inspection. The performance of certain standard compression methods is tested on the test image data. Three types of test images are used for experiments in the thesis. Test image set 1 consist of magnitude reconstructions corresponding to four different cross sections (image 1.5.3). Images in set 2 (image 1.5.4) are obtained following the general methodology proposed by Rathore [62],[37] for sharpening images to improve the visual diagnostic resolution. In the test image set III (image 1.5.5), we consider  $T_1$ ,  $T_2$  and spin density weighted images. These images corresponding to the same cross sections are considered. These images are obtained by using suitable experimental parameters TE, TR. The TE, TR values corresponding to each of these images are given below the images. Performances of standard compression routines are tested for the test images and are shown in tables (1.5.1)-(1.5.3). Huffman coding and Arithmetic coding are techniques which reduces the statistical redundancies considering the data set as one arising from a zero order Markov process. Therefore, the average bit rate achieved through Huffman and Arithmetic coding cannot be less than the entropy of the image data. We see from tables (1.5.1)-(1.5.3) that bit rate achieved by Huffman and Arithmetic coding are quite close to the entropy. JPEG uses transform-coding techniques based on DCT. In case of JPEG compression, redundancy measures based on smoothness (defined more precisely in chapter 3, of the present thesis) are more appropriate. For smooth images we observe the compression is more in case of JPEG. Another compression method, which is widely used, for compression of gray level image is GIF (cf: 1.4.2). It is based on dictionary based coding techniques. Higher the correlation between adjacent pixels, i.e. higher the smoothness of images, methods like JPEG and GIF attains higher compression ratios. The smoothness measures introduced in chapter 3 are consistent with the performance of GIF.

## 1.7 Thesis organization:

Rest of the thesis is organized as follows:

**Chapter II:** is a study of compression of MRI images by a sequence of trigonometric polynomials arising from the Fourier series of the underlying analog intensity distribution

function. The first section concerns with certain preliminaries including the definition of Besov Spaces. These spaces are identified with a weighted sequence space  $l_q^\alpha(X)$ , their norm equivalence is obtained in section two. In section three, we prove a Hardy type inequality and some lemmas including bound on Fourier coefficients by integral modulus of smoothness. These lemmas are used in our main theorem and subsequent storage bounds. Direct and inverse theorems which relates MR images in Besov spaces and their degree of approximation by partial sums of Fourier series are presented in section 4. For the case  $1 < p < \infty$ , approximation by partial sums of Fourier series are used and for the case  $p = 1, \infty$  Steklov approximations are used. Compression algorithm and theoretical storage bounds for MR images are obtained in section five. Section six deals with the experimental implementation of the proposed compression algorithm. In view of some advantages by storing the raw signal data instead of images, we also study the storage of MRI raw data.

**Chapter III:** is a study of compression methods using finite differences. For sufficiently smooth images, higher compression ratios are obtainable through finite difference methods. We store  $k$ -th differences instead of pixel values, the value of  $k$  is determined by the smoothness of the image. In the first section, the method of storing  $k$ -th differences instead of pixel values for compression is discussed. For images belonging to the space  $X^{k,\alpha}$  we estimate the smoothness by measuring the parameter  $\alpha$ . Estimated smoothness of test images, compression algorithm and the corresponding storage requirements are presented in section two. In section three an inductive proof of a lemma due to S.N.Bernstein and certain other inequalities are presented. The reconstruction procedure adopted is approximation by local interpolation by  $k$ -th degree algebraic polynomials. Direct and inverse theorems for the images arising from MRI are obtained in section four. In section five, the experimental implementation of the proposed compression algorithm is carried out on the test images.

**Chapter IV:** is a study of compression by reversible smoothing. Four different approximation process Fejer, Poison, Jackson and Steklov means are used for smoothing the original images. The window functions and the multipliers corresponding to the four

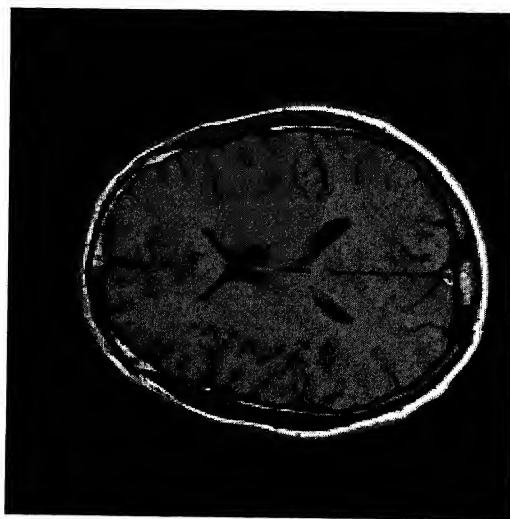


approximation processes are presented in the first section. In section two, compression algorithm and the performance of the approximation processes at different level of smoothing are discussed; they are followed by experimental implementations of the proposed compression strategy. In section three, we obtain a relation connecting depth, resolution and smoothness of a digital image. In section four we obtain storage bound for the compression of images through wavelets. Finally we experimentally compare the performance of wavelet compression methods and the compression strategy based on different approximation processes.

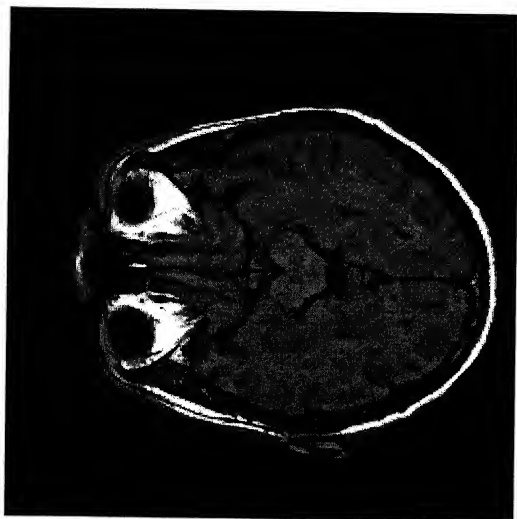
## Magnitude reconstructions -Uncompressed (Test images set 1)



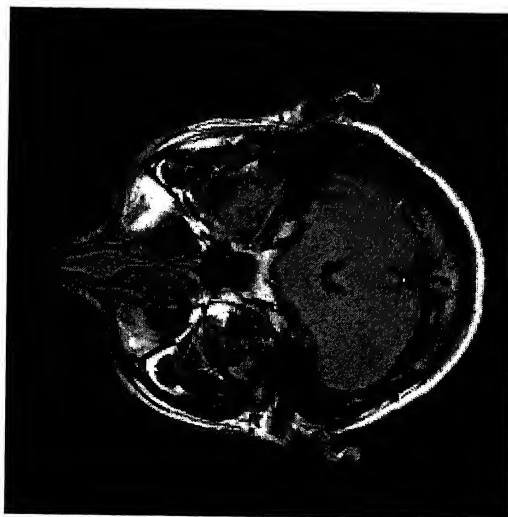
60.img uncompressed



64.img uncompressed



68.img uncompressed



72.img uncompressed

Image 1.5.3

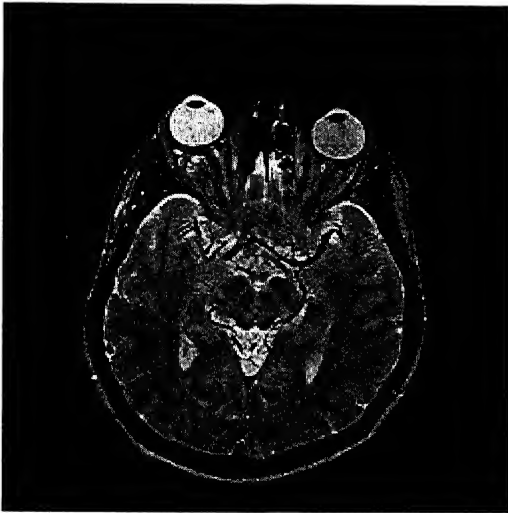
## Sharpened images (Test images set 2)



img.36



img.3612



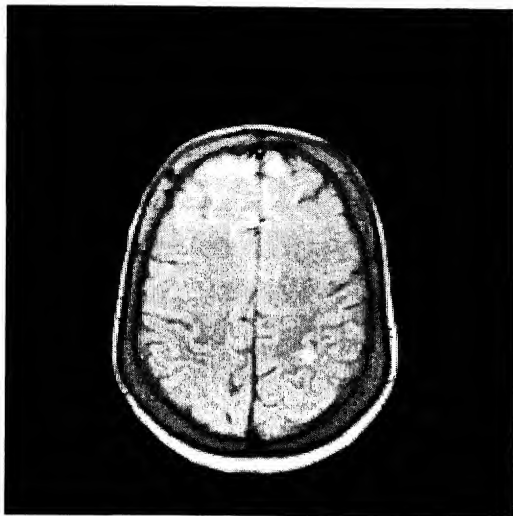
img.36123



img.361234

Image 1.5.4

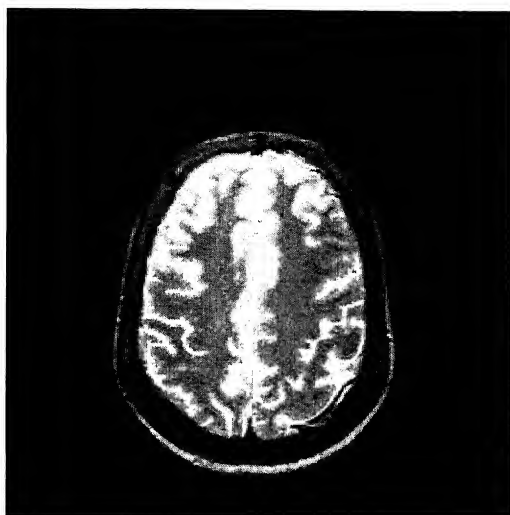
### PD T1 and T2 Weighted Images (Test image set 3)



PD (TE=12 msec, TR=2200 msec)



T1 (TE=14 msec, TR=1000 msec)



T2 (TE=80 msec, TR=2200 msec)

## Compression of test images by standard methods

Image Sizes are measured in Bytes

Percentage Compression = (Original image size – Compressed image size) / (Original image size)

Bit rate = Average number of bits required to code a pixel in an image. It is given as bits per pixel.,

256 X 256 test images are considered. Their initial representation is 8 bits per pixel. Initial Size of image is 65536 bytes.

Here we consider four different compression methods(standards) and tested on the four test images. Size in the table represents size of the compressed image in bytes. P.C represents percentage compression. B.R stands for bit rate in bits per pixel.

In the following tables Size refers to size of the compressed images in bytes.

TEST IMAGE SET 1

Images	Entrop y	HUFFMAN			ARITHMETIC			JPEG			GIF		
		SIZE	P.C	B.R.	SIZE	P.C	B.R	SIZE	P.C	B.R	SIZE	P.C	B.R
60.IMG	5.503	45927	29.92	5.606	45581	30.45	5.564	38266	41.61	4.67	36275	44.65	4.428
64.IMG	5.533	46137	29.60	5.632	45824	30.08	5.594	38195	41.72	4.66	35617	45.65	4.348
68.IMG	5.796	48193	26.56	5.883	47865	29.96	5.843	40202	38.66	4.91	38163	41.77	4.659
72.IMG	5.795	48212	26.43	5.885	47882	26.94	5.845	42027	35.87	5.13	40108	38.80	4.896

Table 1.5.1

# TEST IMAGE SET 2

Images	Entropy y	HUFFMAN				ARITHMETIC				JPEG				GIF			
		SIZE	P.C	B.R.		SIZE	P.C	B.R		SIZE	P.C	B.R		SIZE	P.C	B.R	
36	5.136	47832	27.01	5.839		47741	27.15	5.839		36025	45.03	4.398		41372	36.87	5.052	
3612	5.324	45504	30.57	5.555		44637	31.89	5.449		38750	40.87	4.730		40770	37.39	4.977	
36123	5.369	45175	31.07	5.515		45031	31.29	5.497		40903	37.59	4.993		41455	36.74	5.060	
361234	5.008	43234	34.03	5.278		42405	35.30	5.176		41165	37.19	5.025		40198	38.66	4.907	

Table 1.5.2

# TEST IMAGE SET 3

Images	Entropy	HUFFMAN				ARITHMETIC				JPEG				GIF			
		SIZE	P.C	B.R.		SIZE	P.C	B.R		SIZE	P.C	B.R		SIZE	P.C	B.R	
P.D.	4.9808	42204	35.60	5.152		41524	36.64	5.069		32220	50.84	3.933		30642	53.24	3.740	
T1	4.4895	38525	41.22	4.703		38148	41.79	4.657		25005	61.85	3.052		24562	62.52	2.998	
T2	5.6792	47221	27.95	5.764		46912	28.42	5.727		39149	40.26	4.779		38560	41.16	4.707	

Table 1.5.3

# CHAPTER II: COMPRESSION THROUGH TRIGONOMETRIC POLYNOMIALS

## 2.0 Introduction:

A study of Besov Spaces  $B_q^{\alpha,k}(L_p(I))$  (defined more precisely in section 1 of this chapter) has been made in the context of compression of wavelet decompositions by R.A.DeVore, B.Jawerth and V.A.Popov [17] in 1992. They have characterized functions having certain prescribed degree of approximation by dyadic splines. The interpolation spaces between a pair of Besov Spaces have also been studied by R.A.DeVore and V.A.Popov [16]. In the present chapter, besides characterizing MRI images by a sequence of trigonometric polynomials obtained from Fourier Series of the cross-section  $f$ , we obtain theoretical storage bounds for the proposed compression algorithm.

The remainder of this chapter is organized as follows. The first section concerns with certain preliminaries including the definition of Besov Spaces. These spaces are identified with a weighted sequence space  $l_q^\alpha(X)$  (defined more precisely in section 2), their norm equivalence is obtained in section two. In section three, we prove a Hardy type inequality and some lemmas including bound on Fourier coefficients by integral modulus of smoothness. These lemmas are used in our main theorem and subsequent storage bounds. Direct and inverse theorems, which relate MR images in Besov spaces and their degree of approximation by partial sums of Fourier series, are presented in section four. Compression algorithm and theoretical storage bounds for MR images are obtained in section five. Section six deals with the compression of the raw data and the chapter ends with section seven where the experiments and results are discussed.

We use the following notations throughout this chapter. The domain under consideration is the square,  $I=[0,2\pi]^2$ . Let  $f$  be the analog intensity distribution function of the underlying cross section. It is viewed as a mapping from  $I$  to the interval  $[0,1]$ ,

$$f: [0,2\pi]^2 \rightarrow [0,1].$$

The function  $f(x)$  is of two variables, with  $x = (x_1, x_2) \in I$ . Let  $n = (n_1, n_2) \in Z_2^+$  and  $h = (h_1, h_2) \in \mathcal{R}^2$ . The operations addition and dot product are given by,  $x + h = (x_1+h_1, x_2+h_2)$ ,  $x.h = x_1h_1 + x_2h_2$ ,  $x.n = x_1n_1+x_2n_2$ , for  $t > 0$ ,  $|h| < t \Rightarrow |h_1| < t, |h_2| < t$  and for any positive integer  $k$ ,  $kh = (kh_1, kh_2)$ .

## 2.1 Besov Spaces:

For a fixed  $\alpha > 0$  and an integer  $k > 0$ . The  $k$ -th forward difference with a parameter  $h=(h_1, h_2) \in I$  is given by

$$\Delta_h^0 f(x) := f(x),$$

and for  $j = 1, \dots, k$ , 
$$\Delta_h^j f(x) := \Delta_h^{j-1} f(x+h) - \Delta_h^{j-1} f(x), \quad \text{where } x = (x_1, x_2) \in I.$$

For  $1 \leq p < \infty$ , the  $k$ -th modulus of smoothness in  $L_p(I)$  is defined as

$$\omega_k(f, t)_p = \sup_{|h| \leq t} \left( \int_I |\Delta_h^k f(x)|^p dx \right)^{\frac{1}{p}} \quad (2.1.1)$$

and for  $p = \infty$ ,

$$\omega_k(f, t) = \sup_{|h| \leq t} (|\Delta_h^k f(x)|) \quad (2.1.2)$$

The Besov space  $B_q^{\alpha, k}(L_p(I))$  (see, R.A.DeVore, et al., [17]) norm is defined as

$$\|f\|_{B_q^{\alpha, k}(L_p(I))} := \|f\|_p + \left( \int_0^{2\pi} [t^{-\alpha} \omega_k(f, t)_p]^q \frac{dt}{t} \right)^{\frac{1}{q}} \quad (2.1.3)$$

for  $1 \leq p \leq \infty$ ,  $1 \leq q < \infty$  and for  $q = \infty$ ,

$$\|f\|_{B_q^{\alpha, k}(L_p(I))} := \|f\|_p + \sup_{t>0} [t^{-\alpha} \omega_k(f, t)_p] \quad (2.1.4)$$

$f \in B_q^{\alpha, k}(L_p(I))$  if  $\|f\|_{B_q^{\alpha, k}(L_p(I))} < \infty$



The limits of integral, in the right hand side of (2.1.3) go from 0 to  $\infty$  in the definition used by R.A.DeVore [17]. Since we are dealing with  $2\pi$  periodic functions we let the limits to go from 0 to  $2\pi$ . Although Besov spaces are also defined for  $0 < q < 1$ , we restrict ourselves to the normed spaces, i.e.  $q \geq 1$ . We note that when  $q = \infty$ , we require  $\omega_k(f, t)_p$  to decay at least as fast as  $O(t^\alpha)$  as  $t \rightarrow 0$ , whereas when  $q < \infty$  we require  $\omega_k(f, t)_p$  to decay at a slightly faster rate.

The spaces  $B_q^{\alpha, k}(L_p(I))$  and  $B_q^{\alpha, k'}(L_p(I))$  are related as follows,

$$\begin{aligned} \omega_{k+1}(f, t)_p &\leq 2\omega_k(f, t)_p \\ f \in B_q^{\alpha, k}(L_p(I)) &\Rightarrow f \in B_q^{\alpha, k+1}(L_p(I)). \end{aligned} \quad (2.1.5)$$

It follows by induction that for any  $k' > k$ ,  $f$  will be in  $B_q^{\alpha, k'}(L_p(I))$  if  $f$  is in  $B_q^{\alpha, k}(L_p(I))$ . If both  $k$  and  $k'$  are strictly greater than  $\alpha$ , then  $B_q^{\alpha, k}(L_p(I))$  is equivalent to  $B_q^{\alpha, k'}(L_p(I))$ . This result is presented in the form of a theorem.

**Theorem 2.1.1:** If  $k' > k > \alpha$  then

$$\|\cdot\|_{B_q^{\alpha, k}(L_p(I))} \sim \|\cdot\|_{B_q^{\alpha, k'}(L_p(I))}.$$

To prove this theorem we need three lemmas, which are as follows.

**Lemma 2.1.1:** The following identity for  $k$ -th differences of a function of two variable  $f$ , holds

$$\Delta_{2h}^k f(x) = \sum_{v=0}^k \binom{k}{v} \Delta_h^k f(x + vh).$$

**Proof:** For the case of one variable function the above identity is available (see Timan [71]). Here we prove the above identity for two variables by using induction on  $k$ .

For  $k=1$ ,

$$\begin{aligned}
\Delta_{2h}^k f(x) &= f(x_1 + 2h_1, x_2 + 2h_2) - f(x_1, x_2) \\
&= f(x_1 + 2h_1, x_2 + 2h_2) - f(x_1 + h_1, x_2 + h_2) + f(x_1 + h_1, x_2 + h_2) - f(x_1, x_2) \\
&= \Delta_h^k f(x + h) + \Delta_h^k f(x)
\end{aligned}$$

The identity is true for  $k=1$ , let us assume that it is true for  $k-1$  and we prove for  $k$ .

$$\begin{aligned}
\Delta_{2h}^k f(x) &= \Delta_{2h}^{k-1} f(x + 2h) - \Delta_{2h}^{k-1} f(x) \\
&= \sum_{v=0}^{k-1} \binom{k-1}{v} \left[ \Delta_h^{k-1} f(x + 2h + vh) - \Delta_h^{k-1} f(x + vh) \right] \\
&= \sum_{v=0}^{k-1} \binom{k-1}{v} \left[ \Delta_h^{k-1} f(x + 2h + vh) - \Delta_h^{k-1} f(x + h + vh) + \Delta_h^{k-1} f(x + h + vh) - \Delta_h^{k-1} f(x + vh) \right] \\
&= \sum_{v=0}^{k-1} \binom{k-1}{v} \left[ \Delta_h^k f(x + h + vh) + \Delta_h^k f(x + vh) \right] \\
&= \sum_{v=1}^{k-1} \binom{k-1}{v-1} \left[ \Delta_h^k f(x + vh) \right] + \Delta_h^k f(x + kh) + \Delta_h^k f(x) + \sum_{v=1}^{k-1} \binom{k-1}{v} \left[ \Delta_h^k f(x + vh) \right] \\
&= \sum_{v=1}^{k-1} \binom{k}{v} \left[ \Delta_h^k f(x + vh) \right] + \Delta_h^k f(x + kh) + \Delta_h^k f(x), \quad \text{because } \binom{k-1}{v-1} + \binom{k-1}{v} = \binom{k}{v} \\
&= \sum_{v=0}^k \binom{k}{v} \left[ \Delta_h^k f(x + vh) \right]
\end{aligned}$$

**Lemma 2.1.2:** Let  $1 \leq p \leq \infty$ , and  $f \in L_p(I)$  then,

$$\omega_k(f, t)_p \leq M_k t^k \left\{ \|f\|_p + \int_t^{t+k} \frac{\omega_{k+1}(f, u)_p}{u^{k+1}} du \right\},$$

where the constant  $M_k$  does not depend on  $f$ .

**Proof:** For the case of function of one variable and  $p=\infty$ , the proof is available (Timan, [71, pp.108]), for functions of two variables and for  $1 \leq p < \infty$ , the proof is similar, we just give the outline,

$$\begin{aligned}
\left\| \Delta_{2h}^k f(x) - 2^k \Delta_h^k f(x) \right\|_p &= \left\| \sum_{v=0}^k \binom{k}{v} \left\{ \Delta_h^k f(x + vh) - \Delta_h^k f(x) \right\} \right\|_p \quad [\text{By Lemma 2.1.1}] \\
&\leq \sum_{v=1}^k \binom{k}{v} \left\{ \sum_{\mu=0}^{v-1} \left\| \Delta_h^{k+1} f(x + \mu h) \right\|_p \right\}
\end{aligned}$$

$$\leq k2^{k-1}\omega_{k+1}(f, h)_p.$$

Consider,  $\varepsilon = (\varepsilon_1, \varepsilon_2)$ , putting here, successively  $h=2^m\varepsilon$  ( $m=0,1,\dots,k-1$ ) and taking  $0 \leq \varepsilon_1, \varepsilon_2 \leq t$ . We obtain system of inequalities of the form

$$\left\| \Delta_{2^{m+1}\varepsilon}^k f(x) - 2^k \Delta_{2^m\varepsilon}^k f(x) \right\|_{L_p[0, 2\pi-2kh]} \leq k2^{k-1}\omega_{k+1}(f, 2^m\varepsilon)_p.$$

Multiply both sides by  $2^{-(m+1)k}$  and summing for  $m=0,1,\dots,r-1$  and using the fact that  $\omega_{k+1}(f, u)_p$  is non-decreasing with  $u$ , we get

$$\left\| 2^{-rk} \Delta_{2^r\varepsilon}^k f(x) - \Delta_\varepsilon^k f(x) \right\|_p \leq k^2 t^k \int_t^{2^r t} \frac{\omega_{k+1}(f, u)_p}{u^{k+1}} du,$$

which implies that

$$\left\| \Delta_\varepsilon^k f(x) \right\|_p \leq 2^{-(r-1)k} \|f\|_p + k^2 t^k \int_t^{2^r t} \frac{\omega_{k+1}(f, u)_p}{u^{k+1}} du. \quad (2.1.6).$$

Proceeding as in above with  $f(x)$  replaced by  $\phi(x) = f(2\pi - x)$ , i.e.

$\phi(x_1, x_2) = f(2\pi - x_1, 2\pi - x_2)$ , we obtain

$$\left\| \Delta_\varepsilon^k \phi(x) \right\|_p \leq 2^{-(r-1)k} \|f\|_p + k^2 t^k \int_t^{2^r t} \frac{\omega_{k+1}(f, u)_p}{u^{k+1}} du \quad (2.1.7)$$

$$\begin{aligned} \left( \int_0^{2\pi} |\Delta_\varepsilon^k f(x)|^p dx \right)^{\frac{1}{p}} &\leq \left( \int_0^\pi |\Delta_\varepsilon^k f(x)|^p dx \right)^{\frac{1}{p}} + \left( \int_\pi^{2\pi} |\Delta_\varepsilon^k f(x)|^p dx \right)^{\frac{1}{p}} \\ &= J_1 + J_2 \text{ say.} \end{aligned} \quad (2.1.8)$$

We have,

$$\begin{aligned} J_2 &= \left( \int_\pi^{2\pi} |\Delta_\varepsilon^k f(x)|^p dx \right)^{\frac{1}{p}} = \left( \int_0^\pi |\Delta_\varepsilon^k f(2\pi - y)|^p dy \right)^{\frac{1}{p}} \\ &= \left( \int_0^\pi |\Delta_\varepsilon^k \phi(y)|^p dy \right)^{\frac{1}{p}}. \end{aligned} \quad (2.1.9)$$

Hence from (2.1.6), (2.1.7), (2.1.8) and (2.1.9) it follows that

$$\left\| \Delta_\varepsilon^k f(x) \right\|_p \leq 2 \cdot 2^{-(r-1)k} \|f\|_p + 2 \cdot k^2 t^k \int_t^{2^r t} \frac{\omega_{k+1}(f, u)_p}{u^{k+1}} du. \quad (2.1.10)$$

We choose integer  $r=r(t)$  such that  $\frac{2\pi}{4} \leq 2^r tk \leq \frac{2\pi}{2}$  this gives by (2.1.10) that

$$\left\| \Delta_\varepsilon^k f(x) \right\|_{L_p[0, 2\pi - k\varepsilon]} \leq M_k t^k \left\{ \|f\|_p + \int_t^{\pi/k} \frac{\omega_{k+1}(f, u)_p}{u^{k+1}} du \right\}. \quad (2.1.11)$$

Since  $\varepsilon \leq t$  is arbitrary the result follows from (2.1.11).

Now by using the following lemma, Hardy's inequality (Butzer and Berens [10, pp.199]), in conjunction with the lemma 2.1.2 we will complete a proof of theorem 2.1.1.

**Lemma 2.1.3 (Hardy's inequality):** Let  $\alpha > 0$ ,  $1 \leq q \leq \infty$ . If  $\psi(s)$  is a non-negative measurable function on  $(0, \infty)$  (measurable with respect to the measure  $ds/s$ ), then

$$\left\{ \int_0^\infty \left( t^{-\alpha} \int_0^t \psi(s) \frac{ds}{s} \right)^q \frac{dt}{t} \right\}^{\frac{1}{q}} \leq \frac{1}{\alpha} \left\{ \int_0^\infty \left( s^{-\alpha} \psi(s) \right)^q \frac{ds}{s} \right\}^{\frac{1}{q}} \quad (2.1.12)$$

and

$$\left\{ \int_0^\infty \left( t^{-\alpha} \int_t^\infty \psi(s) \frac{ds}{s} \right)^q \frac{dt}{t} \right\}^{\frac{1}{q}} \leq \frac{1}{\alpha} \left\{ \int_0^\infty \left( s^{-\alpha} \psi(s) \right)^q \frac{ds}{s} \right\}^{\frac{1}{q}} \quad (2.1.13)$$

holds.

**Proof of Theorem 2.1.1:**

$$\omega_{k+1}(f, t)_p \leq 2\omega_k(f, t)$$

$\Rightarrow$

$$\|f\|_{B_q^{\alpha, k+1}(L_p(I))} \leq C \|f\|_{B_q^{\alpha, k}(L_p(I))}$$

Conversely, by Lemma 2.1.2,

$$\begin{aligned} \left( \int_0^{2\pi} \left( t^{-\alpha} \omega_k(f, t)_p \right)^q \frac{dt}{t} \right)^{\frac{1}{q}} &\leq \left( \int_0^{2\pi} \left[ t^{-\alpha} \left( M_k t^k \left\{ \|f\|_p + \int_t^{2\pi/2k} \frac{\omega_{k+1}(f, u)_p}{u^{k+1}} du \right\} \right)^q \right] \frac{dt}{t} \right)^{\frac{1}{q}} \\ &\leq M_k 2^{\frac{q-1}{q}} \left( \|f\|_p^q \int_0^{2\pi} t^{(k-\alpha)q-1} dt + M_k \int_0^{2\pi} \left[ t^{k-\alpha} \int_t^{2\pi} \frac{\omega_{k+1}(f, u)}{u^k} \frac{du}{u} \right]^q \frac{dt}{t} \right)^{\frac{1}{q}} \end{aligned}$$

$$\leq M_{k,\alpha} \|f\|_p + M_k I_1, \quad (2.1.14)$$

$$\text{where, } I_1 = \left( \int_0^{2\pi} \left[ t^{k-\alpha} \int_t^{2\pi} \frac{\omega_{k+1}(f, u)}{u^k} \frac{du}{u} \right]^q \frac{dt}{t} \right)^{\frac{1}{q}}$$

We evaluate  $I_1$  by using Lemma 2.1.3. We define

$$\psi(s) = \begin{cases} s^{-k} \omega_{k+1}(f, s)_p & 0 \leq s \leq 2\pi \\ 0 & s > 2\pi \end{cases}$$

$$\begin{aligned} I_1 &= \left( \int_0^\infty \left[ t^{k-\alpha} \int_t^\infty \psi(s) \frac{ds}{s} \right]^q \frac{dt}{t} \right)^{\frac{1}{q}} \\ &\leq C_{\alpha,k} \left( \int_0^\infty \left( s^{k-\alpha} \psi(s) \right)^q \frac{ds}{s} \right)^{\frac{1}{q}} \quad [\text{By Lemma 2.1.3}] \end{aligned}$$

[Where the constant  $C_{\alpha,k}$  depends only on  $\alpha$  and  $k$ ]

$$= C_{\alpha,k} \left( \int_0^{2\pi} \left( s^{k-\alpha} s^{-k} \omega_{k+1}(f, s)_p \right)^q \frac{ds}{s} \right)^{\frac{1}{q}} = C_{\alpha,k} \left( \int_0^{2\pi} \left( s^{-\alpha} \omega_{k+1}(f, s)_p \right)^q \frac{ds}{s} \right)^{\frac{1}{q}}. \quad (2.1.15)$$

Substituting (2.1.15) in (2.1.14) we get,

$$\|f\|_{B_q^{\alpha,k+1}(L_p(I))} \leq C_1 \|f\|_{B_q^{\alpha,k+1}(L_p(I))}$$

Hence the proof.

In the next section we define certain sequence norm and will show its equivalence with this Besov space norm.

## 2.2 Equivalent Norms:

In the present section we identify Besov Space  $B_q^\alpha(L_p(I))$  norm with a weighted sequence space  $l_q^\alpha$  norm by proving their norm equivalence.

**Definition 2.2.1:**

If  $a = \{a_n\}$  is a sequence whose component functions are in the quasi normed space  $X$ , we use the  $l_q^\alpha(X)$  norms, for  $1 \leq q < \infty$

$$\|a\|_{l_q^\alpha(X)} := \left( \sum_{n=1}^{\infty} \left[ n^\alpha \|a_n\|_X \right]^q \frac{1}{n} \right)^{\frac{1}{q}} \quad (2.2.1)$$

with the usual change to a supremum norm when  $q = \infty$ . When  $\{a_n\}$  is a sequence of real numbers, we replace  $\|a_n\|_X$  by  $|a_n|$  in (2.2.1) and denote the resulting norm by  $\|(a_n)\|_{l_q^\alpha}$ .

**Theorem 2.2.1:** For  $f \in L_p(I)$ ,

$$f \in B_q^{\alpha,r}(L_p(I)) \text{ if and only if } \left\{ \omega_r \left( f, \frac{2\pi}{n} \right)_p \right\}_n \in l_q^\alpha.$$

**Proof:**

Let  $\left\{ \omega_r \left( f, \frac{2\pi}{n} \right)_p \right\}_n \in l_q^\alpha$ . Therefore we have  $\sum_{n=1}^{\infty} \left[ n^\alpha \omega_r \left( f, \frac{2\pi}{n} \right)_p \right]^q \frac{1}{n} < \infty$ .

$$\int_0^{2\pi} [t^{-\alpha} \omega_r(f, t)_p]^q \frac{dt}{t} = \sum_{k=1}^{\infty} \int_{2\pi/k+1}^{2\pi/k} [t^{-\alpha} \omega_r(f, t)_p]^q \frac{dt}{t}, \quad [\omega_r \text{ non decreasing}]$$

$$\leq \frac{1}{(2\pi)^{\alpha q}} \sum_{k=1}^{\infty} (k+1)^{\alpha q} \omega_r \left( f, \frac{2\pi}{k} \right)_p^q \frac{1}{k} \quad \because \int_{2\pi/k+1}^{2\pi/k} \frac{dt}{t} \leq \frac{2\pi}{(k+1)k} \frac{(k+1)}{2\pi} = \frac{1}{k}$$

$$\leq C_{\alpha,q} \sum_{k=1}^{\infty} \left[ k^\alpha \omega_r \left( f, \frac{2\pi}{k} \right)_p \right]^q \frac{1}{k} < \infty \text{ by hypothesis.}$$

Therefore it follows that  $f \in B_q^\alpha(L_p(I))$ .

Conversely, let  $f \in B_q^\alpha(L_p(I))$ ,

$$\begin{aligned} \infty &> \int_0^{2\pi} [t^{-\alpha} \omega_r(f, t)_p]^q \frac{dt}{t} = \sum_{n=1}^{\infty} \int_{2\pi/n+1}^{2\pi/n} [t^{-\alpha} \omega_r(f, t)_p]^q \frac{dt}{t} \\ &\geq \sum_{n=1}^{\infty} \left[ n^\alpha \omega_r \left( f, \frac{2\pi}{n+1} \right)_p \right]^q \frac{1}{n(n+1)} \end{aligned}$$

$$\geq \frac{1}{2^{\alpha q}} \sum_{n=1}^{\infty} \left[ (n+1)^{\alpha} \omega_r \left( f, \frac{2\pi}{n+1} \right)_p \right]^q \frac{1}{(n+1)}.$$

Therefore,  $\left\{ \omega_r \left( f, \frac{2\pi}{n} \right)_p \right\}_n \in l_q^{\alpha}$ . Hence the proof.

In the next section we shall prove some lemmas which will be used along with this norm equivalence to characterize Magnetic Resonance Images in Besov spaces.

### 2.3 Hardy type inequality and bound on Fourier Coefficients:

In this section we shall obtain a bound for Fourier coefficients in terms of modulus of smoothness and in what follows, we shall prove Hardy type inequality which will be used in proving the inverse theorem in the next section.

**Lemma 2.3.1:** Let  $c > 1$ ,  $p > 1$  and  $S_n = a_1 + a_2 + \dots + a_n$ , where  $a_j \geq 0$  then

$$\sum_{n=1}^{\infty} n^{-c} S_n^p \leq K_{p,c} \sum_{n=1}^{\infty} n^{-c} (n a_n)^p,$$

where  $K_{p,c}$  is a constant depending only on  $p$  and  $c$ .

**Proof :**

Let  $S_0 = 0$ . Define  $\phi_n = n^{-c} + (n+1)^{-c} + \dots$  then  $\phi_n < k n^{1-c}$ . This follows from the fact

$$\text{that } n^{-c} + (n+1)^{-c} + \dots < \int_n^{\infty} \frac{dx}{x^c} = \frac{n^{1-c}}{1-c} \quad \text{we take } k = \frac{1}{1-c}.$$

$$\begin{aligned} \sum_1^m n^{-c} S_n^p &= \sum_1^m (\phi_n - \phi_{n+1}) S_n^p \\ &= \sum_1^m \phi_n S_n^p - \sum_1^m \phi_{n+1} S_n^p \\ &= \sum_1^m \phi_n S_n^p - \sum_2^{m+1} \phi_n S_{n-1}^p \\ &\leq \phi_1 S_1^p + \sum_2^m \phi_n (S_n^p - S_{n-1}^p) \end{aligned}$$

$$\begin{aligned}
&\leq \phi_1(S_1^p - S_0^p) + \sum_2^m \phi_n(S_n^p - S_{n-1}^p) \\
&\leq \sum_1^m \phi_n(S_n^p - S_{n-1}^p) \\
&\leq k \sum_1^m n^{1-c} (S_n^p - S_{n-1}^p) \\
&\leq k \sum_1^m n^{1-c} (S_n - S_{n-1}) p S_n^{p-1} \\
&\leq k_{p,c} \sum_1^m n^{1-c} S_n^{p-1} a_n, \quad \text{where } k_{p,c} = \frac{p}{1-c} \\
&\leq k_{p,c} \sum_1^m n^{-\frac{c}{p}} n^{-\frac{c}{p'}} S_n^{p-1} (na_n), \quad \text{where } \frac{1}{p} + \frac{1}{p'} = 1 \\
&= k_{p,c} \sum_1^m n^{-\frac{c}{p}} (na_n) n^{-\frac{c}{p'}} S_n^{\frac{p}{p'}}
\end{aligned}$$

$$\sum_1^m n^{-c} S_n^p \leq k_{p,c} \left( \sum_1^m n^{-c} (na_n)^p \right)^{\frac{1}{p}} \left( \sum_1^m n^{-c} S_n^p \right)^{\frac{1}{p'}}$$

[By Holders inequality]

$$\Rightarrow \left( \sum_1^m n^{-c} S_n^p \right)^{\frac{1}{p}} \leq k_{p,c} \left( \sum_1^m n^{-c} (na_n)^p \right)^{\frac{1}{p}}$$

$$\sum_1^m n^{-c} S_n^p \leq K_{p,c} \sum_1^m n^{-c} (na_n)^p, \quad \text{where } K_{p,c} = \left( \frac{p}{1-c} \right)^p$$

Hence the proof.

**Corollary 2.3.1:**

Let  $c > 1$ ,  $q > 1$  and  $S_n = \sum_{i=1}^n \sum_{j=1}^n a_{ij}$ , where  $a_{ij} \geq 0$  then

$$\sum_{n=1}^{\infty} n^{-c} S_n^q \leq K_{q,c}^1 \sum_{n=1}^{\infty} n^{-c} (n^2 a_{nn})^q,$$

where  $K_{q,c}^1$  is a constant depending only on  $q$  and  $c$ .



**Proof:**

Let  $S_n = P_1^n + \dots + P_n^n$ , where  $P_j^n = \sum_{k=1}^n a_{kj}$

$$\begin{aligned} \sum_{n=1}^{\infty} n^{-c} S_n^q &\leq K_{q,c} \sum_{n=1}^{\infty} n^{-c} (nP_n^n)^q, && \text{applying lemma 2.3.1} \\ &= K_{q,c} \sum_{n=1}^{\infty} n^{-c} T_n^q, && \text{where } T_n = \sum_{k=1}^n n a_{kn} \\ &\leq K_{q,c}^1 \sum_{n=1}^{\infty} n^{-c} (n^2 a_{nn})^q \end{aligned}$$

The following lemma bounds the Fourier coefficients.

**Lemma 2.3.2:** For  $f \in L_p(I)$ ,  $1 \leq p \leq \infty$ , and a given  $k \in \mathbb{Z}^+$  we have for  $n \neq 0$

$$|\hat{f}(n)| \leq C_k \omega_k(f, t)_p,$$

where  $n = (n_1, n_2)$ ,  $t = \frac{\pi}{2(|n_1| + |n_2|)} > 0$ . The constant  $C_k$  depends only on  $k$  and is independent of  $f$ .

**Proof:**

Let us consider,  $I = [0, 2\pi]^2$ ,  $h = (h_1, h_2) \in I$ ,  $n = (n_1, n_2) \in \mathbb{Z}_2^+$ ,  $x = (x_1, x_2) \in I$ ,  $dx = dx_1 dx_2$ , and the dot product  $n \cdot x = n_1 x_1 + n_2 x_2$ . We have,

$$\begin{aligned} (\Delta_h^k f)^\wedge(n) &= \frac{1}{4\pi^2} \int_I (\Delta_h^{k-1} f(x+h) - \Delta_h^{k-1} f(x)) e^{-in \cdot x} dx \\ &= \frac{1}{4\pi^2} e^{in \cdot h} \int_I \Delta_h^{k-1} f(x) e^{-in \cdot x} dx - \frac{1}{4\pi^2} \int_I \Delta_h^{k-1} f(x) e^{-in \cdot x} dx \\ &= \frac{1}{4\pi^2} (e^{in \cdot h} - 1) \int_I \Delta_h^{k-1} f(x) e^{-in \cdot x} dx \\ &= \frac{1}{4\pi^2} (e^{in \cdot h} - 1)^k \int_I f(x) e^{-in \cdot x} dx \\ &\Rightarrow \left[ e^{\frac{in \cdot h}{2}} \left( e^{\frac{in \cdot h}{2}} - e^{-\frac{in \cdot h}{2}} \right) \right]^k |\hat{f}(n)| \leq \frac{1}{4\pi^2} \int_0^{2\pi} \int_0^{2\pi} |\Delta_h^k f(x)| dx \end{aligned}$$

$$\Rightarrow \left| 2 \sin \frac{n \cdot h}{2} \right|^k \left| \hat{f}(n) \right| \leq \frac{1}{4\pi^2} (4\pi^2)^{\frac{1}{p}} \left( \int_1 \left| \Delta_h^k f(x) \right|^p \right)^{\frac{1}{p}} \quad [\text{By, Holder's inequality}]$$

Let  $t = \frac{\pi}{2(|n_1| + |n_2|)}$ , and  $h = (h_1, h_2) = \left( \frac{\pi}{2(n_1 + n_2)}, \frac{\pi}{2(n_1 + n_2)} \right)$ . If  $n_1 < 0$ , then we take  $h_1 = -h_1$  and if  $n_2 < 0$ ,  $h_2 = -h_2$ .

$$\Rightarrow 2^k \left| \hat{f}(n) \right| \leq C_p \omega_k(f, t), \quad \text{where } C_p = \frac{1}{(4\pi^2)^{\frac{1}{p}}}$$

$$\Rightarrow \left| \hat{f}(n) \right| \leq C_{p,k} \omega_k(f, t), \quad \text{where } C_{p,k} = \frac{1}{2^k (4\pi^2)^{\frac{1}{p}}}$$

Hence the proof.

In the next section we shall present the direct and inverse theorem.

## 2.4 Direct and Inverse Theorems:

Here for the direct theorem we deal the cases  $1 < p < \infty$  and  $p = 1, \infty$  separately. For the case  $1 < p < \infty$  we use partial sums of Fourier series and for  $p = 1, \infty$  Steklov approximations are used to establish the results. Inverse results are obtained for the best approximation trigonometric polynomials. To prove our main theorem, certain results about trigonometric polynomials of best approximation for functions of two variables are required, and these are available in the literature. Here we just state the results. These results are based on partial modulus of smoothness (Timan, [71]). For functions of two variables the partial modulus of smoothness is defined as follows.

### Definition 2.4.1 (Partial Modulus of Smoothness):

$$\hat{\omega}_k(f; u, 0) = \sup_y \sup_{|t| \leq u} \left| \sum_{v=0}^k (-1)^{k-v} \binom{k}{v} f(x + vt, y) \right|$$

$$\hat{\omega}_k(f; 0, v) = \sup_x \sup_{|t| \leq v} \left| \sum_{v=0}^k (-1)^{k-v} \binom{k}{v} f(x, y + vt) \right|$$

and for  $1 \leq p < \infty$ ,

$$\hat{\omega}_k(f; u, 0)_p^p = \sup_{|t| \leq u} \int_1^t \left| \sum_{v=0}^k (-1)^{k-v} \binom{k}{v} f(x_1 + vt, x_2) \right|^p dx$$

$$\hat{\omega}_k(f; 0, v)_p^p = \sup_{|t| \leq v} \int_1^t \left| \sum_{v=0}^k (-1)^{k-v} \binom{k}{v} f(x_1 + vt, x_2) \right|^p dx$$

**Definition 2.4.2 (Total Modulus of Smoothness):**

$$\hat{\omega}_k(f; u, v) = \sup_{|h_1| < u} \sup_{|h_2| \leq v} \left| \sum_{v=0}^k (-1)^{k-v} \binom{k}{v} f(x_1 + vh_1, x_2 + vh_2) \right|$$

and for  $1 \leq p < \infty$ ,

$$\hat{\omega}_k(f; u, v)_p^p = \sup_{|h_1| < u} \sup_{|h_2| \leq v} \int_1^t \left| \sum_{v=0}^k (-1)^{k-v} \binom{k}{v} f(x_1 + vh_1, x_2 + vh_2) \right|^p dx$$

**Definition 2.4.3 (Mixed Modulus of Smoothness):**

$$\begin{aligned} \hat{\omega}_{k,l}(f; u, v) &= \sup_{|h_1| < u} \sup_{|h_2| \leq v} \left| \Delta_{h_1}^k \Delta_{h_2}^l f(x_1, x_2) \right| \\ &= \sup_{|h_1| < u} \sup_{|h_2| \leq v} \left| \sum_{v=0}^k \sum_{\mu=0}^l (-1)^{k+l-v-\mu} \binom{k}{v} \binom{l}{\mu} f(x_1 + vh_1, x_2 + \mu h_2) \right| \end{aligned}$$

and for  $1 \leq p < \infty$ ,

$$\hat{\omega}_{k,l}(f; u, v)_p^p = \sup_{|h_1| < u} \sup_{|h_2| \leq v} \int_1^t \left| \sum_{v=0}^k \sum_{\mu=0}^l (-1)^{k+l-v-\mu} \binom{k}{v} \binom{l}{\mu} f(x_1 + vh_1, x_2 + \mu h_2) \right|^p dx$$

The relation between moduli of smoothness is as follows (Timan, [71]).

$$\max\{\hat{\omega}_k(f; u, 0), \hat{\omega}_k(f; 0, v)\} \leq \hat{\omega}_k(f; u, v) \leq \sum_{v=0}^k \binom{k}{v} \hat{\omega}_{k-v,v}(f; u, v) \quad (2.4.1)$$

For the case when  $u = v = t$ , we have  $\hat{\omega}_k(f; u, v) = \omega_k(f; t)$ , where  $\omega_k(f; t)$  for functions of two variable is defined by equations (2.1.1) and (2.1.2). Let  $E_n^*(f)_p = \|f - T_n\|_p$  where the function  $f$  is of two variables, and  $T_n(x)$  is the trigonometric polynomial of best approximation, or order  $n, n$  in the corresponding variables  $x_1, x_2$ .

**Theorem 2.4.1:** Consider the function  $f(x)$  with  $x = (x_1, x_2)$ , defined throughout the whole space of the variables  $x_1$  and  $x_2$  and having period  $2\pi$  in each of these variables. If  $1 \leq p \leq \infty$ , and if  $f \in L_p(I)$ , then for any integers  $k_1, k_2, n \geq 0$ , with  $k = (k_1, k_2)$  we have

$$E_n^*(f)_p \leq C_k \left( \hat{\omega}_{k_1} \left( f; \frac{1}{n+1}, 0 \right)_p + \hat{\omega}_{k_2} \left( f; 0, \frac{1}{n+1} \right)_p \right),$$

where  $C_k$  is a constant depending only on  $k$ , and  $\omega_k(f; t_1, t_2)_p$  is the corresponding integral partial modulus of smoothness.

From the above theorem and (2.4.1) it is clear that, for  $1 \leq p \leq \infty$ ,

$$E_n^*(f)_p \leq C_k \omega_k(f, t)_p, \quad (2.4.2)$$

where  $t = \frac{1}{n+1}$  and  $k = k_1 = k_2$ . The  $\omega_k(f; t)$  in the right hand side of (2.4.2) is the modulus of smoothness for functions of two variables, as defined in (2.1.1). Next we state a theorem which relates approximation by partial sums of Fourier series and the best approximation in case of  $1 < q < \infty$ .

**Theorem 2.4.2:** If  $1 < q < \infty$  and if the function  $f(x)$ ,  $x = (x_1, x_2)$ , periodic in both the variables of period  $2\pi$  belongs to  $L_q$  on  $I$ , then for any  $n \geq 0$ ,

$$\left\{ \int_I |f(x) - S_n^*(f; x)|^q dx \right\}^{\frac{1}{q}} \leq C_q E_n^*(f)_q,$$

where  $C_q$  is a constant depending only on  $q$  and  $S_n^*(f; x)$  is the partial sum of order  $n, n$  of the Fourier Series of  $f(x_1, x_2)$ .

**Proof:**

Let  $T_n(x)$  be the trigonometric polynomial of best approximation, of order  $n, n$  in the corresponding variables  $x_1, x_2$ . We have,

$$\begin{aligned}
\|f - S_n^*(f)\|_q &= \|f - T_n + T_n - S_n^*(f)\|_q \leq (1 + \|S_n^*\|_q) \|f - T_n\|_q \\
&\leq C_q \|f - T_n\|_q \quad S_n^* \text{ is a bounded operator in } L_q(I) \text{ for } 1 < q < \infty \\
&\quad \text{and the constant } C_q \text{ depends only on } q \\
&\leq C_q E_n^*(f)_q
\end{aligned}$$

Next we state the converse theorem for best approximations by trigonometric polynomials.

**Theorem 2.4.3:** If for a certain  $q$  ( $1 \leq q \leq \infty$ ) the function  $f(x)$  belongs to the class  $L_q$  over the square  $I$ , and  $E_{n_1, n_2}^*(f)$  ( $n=0, 1, 2, \dots$ ) is the sequence of best approximation by trigonometric polynomials  $T_{n_1, n_2}(x)$ , of order  $n_1, n_2$  in the corresponding variables  $x_1$  and  $x_2$ , then whatever the integers  $k_1, k_2$  may be, the inequality

$$\hat{\omega}_{k_1, k_2} \left( f, \frac{1}{n}, \frac{1}{n} \right)_q \leq \frac{C_{k_1, k_2}}{n^{k_1 + k_2}} \sum_{v_1=0}^n \sum_{v_2=0}^n (v_1 + 1)^{k_1 - 1} (v_2 + 1)^{k_2 - 1} E_{v_1, v_2}^*(f)_q$$

remains valid, where  $C_{k_1, k_2}$  is a constant depending only on  $k_1, k_2$  and  $\hat{\omega}_{k_1, k_2} \left( f, \frac{1}{n}, \frac{1}{n} \right)_q$

is the corresponding mixed modulus of smoothness.

**Theorem 2.4.4:** If  $f \in B_q^{\alpha, k}(L_p(I))$  for a certain  $p$  ( $1 < p < \infty$ ),  $q$  ( $1 \leq q < \infty$ ) and for any  $k$  and  $\alpha$  such that  $0 < \alpha < k$ , then  $\{E_n(f)_p\}_n \in l_q^\alpha$  where  $E_n(f)_p = \|f - S_n(f)\|_p$  and  $S_n(f)$  is the partial sum of order  $n$ ,  $n$  in the corresponding variables  $x_1, x_2$ , of the Fourier series for the function  $f$ .

**Proof:**

$$\sum_{n=1}^{\infty} [n^\alpha E_n(f)_p]^q \frac{1}{n} \leq C_q \sum_{n=1}^{\infty} [n^\alpha E_n^*(f)_p]^q \frac{1}{n} \quad [\text{By theorem 2.4.2}]$$

$$\leq C_{q, k} \sum_{n=1}^{\infty} \left[ n^\alpha \omega_k \left( f, \frac{1}{n} \right)_p \right]^q \frac{1}{n} \quad [\text{By (2.4.2)}]$$

$$< \infty$$

[By theorem 2.2.1]

$\Rightarrow \{E_n(f)_p\}_n \in l_q^\alpha$ . Hence the proof.

For the case  $p = 1, \infty$  we make use of Steklov approximations

**Definition 2.4.4 ( $W^{m,p}(I)$  Space):**

Let  $1 \leq p \leq \infty$ ,  $W^{m,p}(I)$  is space of all functions for which  $\|f\|_{m,p} < \infty$ , where the norm is given by,

$$\|f\|_{m,p} = \|f\|_{L_p(I)} + \|f^{(m)}\|_{L_p(I)}$$

**Steklov Approximations:**

**Definition 2.4.5:**

The  $m$ -th Steklov approximations  $\{f_{\eta,m}\}$ , ( $\eta > 0$ ,  $m = 1, 2, \dots$ ) is defined as

$$f_{\eta,m}(x) = \left(\frac{\eta}{m}\right)^{-m} \int_0^{\frac{\eta}{m}} \dots \int_0^{\frac{\eta}{m}} \left\{ f(x) - (-1)^m \Delta_{\sum_{i=1}^m \eta_i}^m f(x) \right\} d\eta_1 \dots d\eta_m.$$

For  $1 \leq p \leq \infty$ , for every  $f \in L_p(I)$ ,  $\{f_{\eta,m}\} \in W^{m,p}(I)$  and we have the following:

**Theorem 2.4.5:** For every  $f \in L_p(I)$   $1 \leq p \leq \infty$  and  $\eta > 0$ ,

$$\|f - f_{\eta,m}\|_p \leq \omega_m(f, \eta)_p.$$

**Proof :**

$$f_{\eta,m}(x) - f(x) = \left(\frac{\eta}{m}\right)^{-m} \int_0^{\frac{\eta}{m}} \dots \int_0^{\frac{\eta}{m}} \left\{ (-1)^{m+1} \Delta_{\sum_{i=1}^m \eta_i}^m f(x) \right\} d\eta_1 \dots d\eta_m$$

Using Generalized Minkowski's inequality we have,

$$\begin{aligned} \|f_{\eta,m} - f\|_p &\leq \left(\frac{\eta}{m}\right)^{-m} \left(\frac{\eta}{m}\right)^m \left\| \Delta_{\sum_{i=1}^m \eta_i}^m f(x) \right\|_p \\ &\leq \sup_{|h| < \eta} \left\| \Delta_h^m f(x) \right\|_p \\ &= \omega_m(f, \eta)_p. \end{aligned}$$

**Theorem 2.4.6:** If  $f \in B_q^{\alpha,k}(L_p(I))$  for a certain  $p$  ( $1 \leq p \leq \infty$ ),  $q$  ( $1 \leq q < \infty$ ) and for any  $k$  and  $\alpha$  such that  $0 < \alpha < k$ , then  $\{e_n(f)_p\}_n \in l_q^\alpha$  where  $e_n(f)_p = \|f - f_n\|_p$  and  $f_n$ 's correspond to  $f_{\eta,k}$ , the  $k$ -th Steklov approximation of the function  $f$ , for  $\eta = \frac{k}{n}$ .

**Proof:**

The proof of theorem 2.4.6 is similar to that of theorem 2.4.4. The proof for this case uses theorem 2.4.5.

Now we shall state and prove the inverse theorem. Inverse theorems are proved for the best approximation trigonometric polynomials.

**Theorem 2.4.7:** If for a certain  $p$  ( $1 \leq p \leq \infty$ ) the function  $f \in L_p(I)$  and  $\{E_n^*(f)_p\}_n \in l_q^\alpha$  where  $E_n^*(f)_p = \|f - T_n\|_p$  and  $T_n(x)$  with  $x=(x_1, x_2)$  is the best approximation trigonometric polynomial, of order  $n$ ,  $n$  in the corresponding variables  $x_1, x_2$ , for the function  $f$ , then for all  $k$ , and  $\alpha$  such that  $0 < \alpha < k$ , the function  $f \in B_q^{\alpha,k}(L_p(I))$ .

**Proof :**

If, we show that,  $\sum_{n=1}^{\infty} \left[ n^\alpha \omega_k\left(f, \frac{1}{n}\right)_p \right]^q \frac{1}{n} < \infty$ , then the above theorem follows from the theorem 2.2.1.

Consider,  $\hat{\omega}_k$  and  $\hat{\omega}_{k,v}$  the total and mixed modulus of smoothness respectively (cf: definitions 2.4.2 and 2.4.3). For  $n > 0$ , we have,

$$\begin{aligned} \omega_k\left(f; \frac{1}{n}\right) &= \hat{\omega}_k\left(f; \frac{1}{n}, \frac{1}{n}\right) \leq \sum_{v=0}^k \binom{k}{v} \hat{\omega}_{k-v,v}\left(f; \frac{1}{n}, \frac{1}{n}\right) \\ &\leq \sum_{v=0}^k \binom{k}{v} \frac{C_k}{n^k} \sum_{v_1=0}^n \sum_{v_2=0}^n (v_1+1)^{k-v-1} (v_2+1)^{v-1} E_{v_2 v_3}^*(f) \quad [\text{By theorem 2.4.3}] \\ &\leq \sum_{v=0}^k \binom{k}{v} \frac{C_k}{n^k} \sum_{v_1=0}^n \sum_{v_2=0}^n n^{k-2} E_{v_2 v_3}^*(f) \end{aligned}$$

$$\leq 2^k k \frac{C_k}{n^k} \sum_{v_1=0}^n \sum_{v_2=0}^n n^{k-2} E_{v_2 v_3}^*(f)$$

[where  $E_{v_1 v_2}^*(f)$  is the best approximation of order  $v_1, v_2$  in the

corresponding variables  $x_1, x_2$  of  $f(x)$ .]

Therefore, we have

$$\sum_{n=1}^{\infty} \left[ n^{\alpha} \omega_k \left( f, \frac{1}{n} \right)_p \right]^q \frac{1}{n} \leq \sum_{n=1}^{\infty} \left[ n^{\alpha} \frac{C_k}{n^k} \sum_{v_1=1}^n \sum_{v_2=1}^n n^{k-2} E_{v_2 v_3}^*(f)_p \right]^q \frac{1}{n}$$

$$\leq C_k \sum_{n=1}^{\infty} n^{\alpha q - kq - 1} S_n^q$$

$$[\text{where } S_n = \sum_{i=1}^n \sum_{j=1}^n a_{ij} \quad \text{and} \quad a_{ij} = n^{k-2} E_{ij}^*(f)_p]$$

$$\leq C_k \sum_{n=1}^{\infty} n^{-(1+(k-\alpha)q)} S_n^q$$

$$[(1 + (k-\alpha)q) > 1 \because q > 0 \text{ and } 0 < \alpha < k]$$

Applying the corollary of lemma 2.3.1 we get,

$$\begin{aligned} &\leq C_{q,k} \sum_{n=1}^{\infty} n^{-(1+(k-\alpha)q)} (n^2 a_{nn})^q \\ &\leq C_{q,k} \sum_{n=1}^{\infty} [n^{\alpha} E_n^*(f)_p]^q n^{-1} n^{-kq} n^{kq} \\ &\leq C_{q,k} \sum_{n=1}^{\infty} [n^{\alpha} E_n^*(f)_p]^q \frac{1}{n} \\ &< \infty, \end{aligned}$$

by hypothesis.

Here the constant  $C_{q,k}$  depends only on  $q$  and  $k$  and is independent of the function  $f$ .

$\Rightarrow f \in B_q^{\alpha,k}(L_p(I))$  and hence the proof.

In the next section we propose a compression algorithm for MR images and we obtain storage bounds.



## 2.5 Compression Algorithm and Storage bounds:

Let the given MRI image be

$$f(x, y) = \sum_{j,k} c_{j,k} e^{i(jx+ky)}. \quad (2.5.1)$$

We apply an invertible transformation on the Fourier coefficient's to get a new function  $\tilde{f}$ . This transformation could be based on a known convolving function or otherwise, so that

$$\tilde{f}(x, y) = \sum_{j,k} \tilde{c}_{j,k} e^{i(jx+ky)} \quad (2.5.2)$$

with  $|\tilde{c}_{j,k}| \leq |c_{j,k}|$  holding. The next step in the compression algorithm is to modify  $\tilde{c}_{j,k}$ 's to  $c_{j,k}^*$ 's such that only non zero  $c_{j,k}^*$ 's are to be retained. Normally the zero coefficients occur in clusters so that the storage indexing is a minimal. The Compression Ratio (C.R) is given by

$$C.R = \frac{\text{Uncompressed image size} - \text{Compressed image size}}{\text{Uncompressed image size}}. \quad (2.5.3)$$

A higher compression ratio in the method is obtained indeed because of the number of vanishing  $c_{j,k}^*$ 's. The compressed image data file contains a header. Patient information and certain parameters are stored in the header. We store the zero frequency term in the header and all other frequency terms are stored by using the following algorithm.

### Algorithm 2.5.1 :

**Step 1 :** Calculate  $c_{j,k}$ 's from the pixel data  $p_{j,k}$ 's using FFT.

**Step 2 :** Calculate  $\tilde{c}_{j,k}$ 's from  $c_{j,k}$ 's for the chosen convolving function.

**Step 3 :** Choose a positive integer  $N$  and numbers  $1 < p < \infty$ ,  $\alpha > 0$ . Choose  $q$  such that  $\alpha q > 2$  and quantized coefficients  $c_{j,k}^*$  that satisfy

$$(i) |\tilde{c}_{j,k} - c_{j,k}^*|^q \leq \frac{1}{N},$$

$$(ii) |\tilde{c}_{j,k}| \leq |c_{j,k}| \text{ and}$$

$$(iii) \tilde{c}_{j,k} = 0 \text{ if } |\tilde{c}_{j,k}| \leq \frac{1}{N}$$

**Step 4:** Store non zero  $c_{j,k}^*$ 's by a variable length coding procedure (cf: 1.2).

Now we give a bound on storage space required for the above algorithm, here the  $\tilde{c}_{j,k}$ 's are the Fourier coefficients of the pixel data and  $\tilde{f}$  is the approximation of  $f$  by partial sums of Fourier series.

**Definition 2.5.1:**

Let  $n = (n_1, n_2)$

$$\hat{f}_s^m(|n|) = \max \left\{ |\hat{f}(n_1, n_2)| : |n_1| + |n_2| = |n| \right\}.$$

From the above definition it is clear that,

$$\sum_{n_1 \in \mathbb{Z}} \sum_{n_2 \in \mathbb{Z}} |\hat{f}(n_1, n_2)| \leq \sum_{|n|=1}^{\infty} 4 |n| \hat{f}_s^m(|n|) + \hat{f}_s^m(0), \quad (2.5.4)$$

where  $\hat{f}_s^m(0)$  is the maximum of the set containing one element,  $\hat{f}(0)$ . It is stored in the header. We do not include the zero frequency information, in our compression estimates. A small storage space is usually reserved for the header information.

By lemma 2.3.2, we have,

$$|\hat{f}_s^m(|n|)| \leq C_k \omega_k \left( f; \frac{1}{|n|} \right). \quad (2.5.5)$$

The above relation is used in obtaining storage bounds for the algorithm 2.5.1

**Theorem 2.5.1 (Storage bound):**

If  $f \in B_q^{\alpha,k}(L_p(I))$  for  $1 < p < \infty$ ,  $0 < q \leq \infty$ ,  $0 < \alpha < k$  and  $\alpha q > 2$ , then  $\eta$  the number of non zero coefficients  $c_{j,k}^*$ 's in the above algorithm satisfies,

$$\eta \leq C_k N |f|_{B_q^{\alpha,k}(L_p(I))}^q,$$

where  $|f|_{B_q^{\alpha,k}(L_p(I))} = \left( \int_0^{2\pi} \left[ t^{-\alpha} \omega_k(f, t)_p \right]^q \frac{dt}{t} \right)^{\frac{1}{q}}$ .

**Proof:**

Let  $N$  be the quantization parameter in the algorithm 2.5.1 and  $\eta$  is the number of non-zero coefficients. Let the zero frequency information  $\hat{f}(0)$ , be stored in the header as per our compression procedure. We have

$$\frac{\eta}{N} \leq \sum_{\text{non zero } c_{j,k}} |c_{j,k}^*|^q \leq \sum_{j,k} |c_{j,k}|^q \leq \sum_{n_1 \in \mathbb{Z}} \sum_{n_2 \in \mathbb{Z}} |\hat{f}(n_1, n_2)|^q \quad [\because |c_{j,k}^*| \leq |c_{j,k}| \text{ by algorithm 2.5.1}]$$

$$c_{j,k} = \hat{f}(j, k)$$

$$\leq \sum_{|n|=1}^{\infty} 4 |n| |\hat{f}_s^m(|n|)|^q \quad [\text{By (2.5.4)}]$$

[ For  $|n| = 0$ , storage will be in the header]

$$\leq C \sum_{|n|=1}^{\infty} 4 |n| \omega_k \left( f; \frac{1}{|n|} \right)^q \quad [\text{By (2.5.5)}]$$

$$\leq C \sum_{|n|=1}^{\infty} |n|^{\alpha q - 1} \omega_k \left( f; \frac{1}{|n|} \right)^q \quad [\because \alpha q > 2 \Rightarrow \alpha q - 1 > 1]$$

$$\leq C \sum_{|n|=1}^{\infty} \left[ |n|^{\alpha} \omega_k \left( f; \frac{1}{|n|} \right) \right]^q \frac{1}{|n|}$$

$$\leq C |f|_{B_q^{\alpha,k}(L_p(I))}^q$$

$$\Rightarrow \eta \leq C_k N |f|_{B_q^{\alpha,k}(L_p(I))}^q$$

**Poisson Approximation Process:**

Poisson kernel (see Zygmund [79]), is convolved with the pixel data to obtain Poisson's approximation for the image. This is achieved by multiplying the Fourier coefficients  $\hat{f}(m, n)$  by  $r^{|m|+|n|}$ , for  $0 < r < 1$ . This convolution imposes smoothness in the image. Lower the value of  $r$  higher the smoothness. The value of  $r$  is usually taken to be near 1 to avoid blurring. Algorithm 2.5.1 is applied for image compression with coefficients  $\tilde{c}_{j,k}$  as the Poisson coefficients,

i.e.  $\tilde{c}_{m,n} = \hat{f}(m,n)r^{|m|+|n|}$ . Let the Poisson filtered image be represented as,

$$\tilde{f}(x,y) = \sum_{j,k} \tilde{c}_{j,k} e^{i(jx+ky)}. \quad (2.5.6)$$

Now the quantized coefficients  $c_{j,k}^*$  's are chosen as per algorithm 2.5.1.

$$|\tilde{c}_{j,k} - c_{j,k}^*|^q \leq \frac{1}{N},$$

$$|r^{|j|+|k|}\tilde{c}_{j,k}|^q = |c_{j,k}^*|^q \leq \frac{1}{N} \Rightarrow c_{j,k}^* = 0.$$

The compressed image is represented as,

$$f^*(x,y) = \sum_{j,k} c_{j,k}^* e^{i(jx+ky)}. \quad (2.5.7)$$

**Theorem 2.5.2 (Storage bound for Poisson approximation process):**

If  $\tilde{f} \in B_q^{\alpha,k}(L_p(I))$  for  $1 < p < \infty$ ,  $0 < q \leq \infty$ ,  $0 < \alpha < k$ ,  $0 < r < 1$ , and  $\alpha q > 2$ , then  $\eta$  the number of non zero coefficients  $c_{j,k}^*$  's in the above algorithm satisfies,

$$\eta \leq C_k N r^q \left| \tilde{f} \right|_{B_q^{\alpha,k}(L_p(I))}^q,$$

$$\text{where } \left| \tilde{f} \right|_{B_q^{\alpha,k}(L_p(I))} = \left( \int_0^{2\pi} \left[ t^{-\alpha} \omega_k(\tilde{f}, t)_p \right]^q \frac{dt}{t} \right)^{\frac{1}{q}}.$$

**Proof:**

Let  $N$  be the quantization parameter as in the algorithm 2.5.1 and  $\eta$  is the number of non zero coefficients, we have

$$\frac{\eta}{N} \leq \sum_{\text{non zero } c_{j,k}^*} |c_{j,k}^*|^q \leq \sum_{j,k} |c_{j,k}^*|^q \leq \sum_{n_1 \in \mathbb{Z}} \sum_{n_2 \in \mathbb{Z}} |r^{|n|} \hat{f}(n_1, n_2)|^q \quad [\because |c_{j,k}^*| \leq |c_{j,k}| \text{ by algorithm 2.5.4}]$$

$$c_{j,k} = \hat{f}(j,k) \text{ and } \hat{f}(0,0) \text{ is stored in header}]$$

$$\leq \sum_{|n|=1}^{\infty} 4 |n| \left| r^{|n|} \hat{f}_s^m(|n|) \right|^q \quad [\text{By (2.5.3)}]$$

$$\leq C r^q \sum_{|n|=1}^{\infty} 4 |n| \omega_k \left( f; \frac{1}{|n|} \right)^q \quad [\text{By (2.5.4) and } 0 < r < 1]$$

$$\begin{aligned}
&\leq C_1 r^q \sum_{|n|=1}^{\infty} |n|^{\alpha q - 1} \omega_k \left( f; \frac{1}{|n|} \right)^q && [\because \alpha q > 2 \Rightarrow \alpha q - 1 > 1] \\
&\leq C_1 r^q \sum_{|n|=1}^{\infty} \left[ |n|^{\alpha} \omega_k \left( f; \frac{1}{|n|} \right) \right]^q \frac{1}{|n|} \\
&\leq C_k r^q |f|_{B_q^{\alpha,k}(L_p(I))}^q
\end{aligned}$$

$$\Rightarrow \eta \leq C_k N r^q |f|_{B_q^{\alpha,k}(L_p(I))}^q.$$

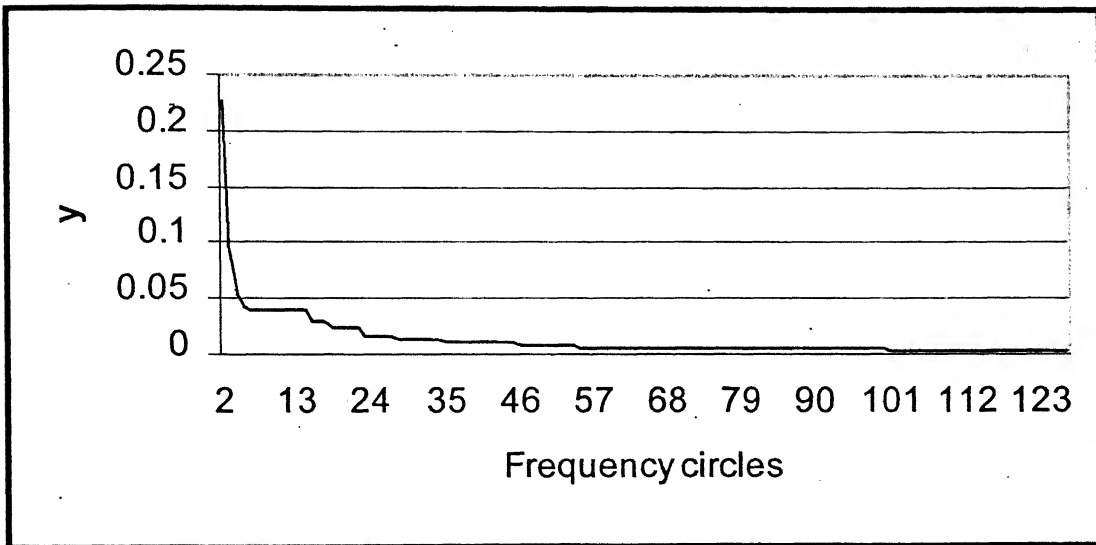
## 2.6 Compression of Raw Data:

The usual MRI image refers to the magnitude image reconstruction from the complex signal data. However, MRI offers various kinds of reconstructions like real, imaginary, phase, other than the usual magnitude images, and each one give information of specific significance for medical diagnosis. For example, the phase images (see Stark and Bradley [69]) help determine if a blood vessel is patent or if it is filled with clotted blood instead. The presence of flowing blood is suggested if a blood vessel in a phase image has a pattern of stripes distinct from tissues that are known to be stationary. They also provide information about the flow of Cerebro Spinal Fluid (CSF) (See Partain et al., [55]). Real images are advantageous for certain techniques notably inversion recovery (See Parain et al., [56]). The information present in real, imaginary and phase images thus complement the strength of MR signals present in the modulus images. It is possible to reconstruct all these images if the raw data (MR signal) is stored instead of just the magnitude images.

The raw data contains frequency information of the corresponding image. It satisfies decay properties, i.e. the magnitude of the data decays as the frequency increases. Consider concentric circles around the zero frequency term. The maximum magnitude of the raw data in the exterior of the circle can be shown to decay, and the corresponding decay could be faster if the image under investigation is sufficiently smooth. Graph (2.6.1), shows that for the MR images under consideration the maximum magnitude of the raw data in the exterior of the concentric circles centered at zero frequency, decreases

as the radius of the circle increases. This decay property gives rise to compression of raw data. In Graph 2.6.1 Frequency circles in the x-axis refers to the radius of the concentric circles considered with center as the zero frequency term. Instead of storing the raw data in the matrix form, they can be stored in concentric circles and a variable length coding based on the apriori information obtained from graph 2.6.1 can be used so that the preceding zero entries corresponding to each circle need not be stored.

### Decay of Fourier Coefficients



y – Maximum magnitude of the raw data in the exterior of frequency circle.

Graph 2.6.1

## 2.7 Experimental Results:

Test image set 1 consists of magnitude reconstructions corresponding to four different cross sections. The sizes of test images are 256x256. Algorithm 2.5.1 has been implemented on the test image set 1. The required Fourier coefficients are computed using FFT routines, due to Cooley and Tuckey (see Brigham, [8]). Images obtained after certain compression and decompression of test images are shown in Image 2.7.1-2.7.3. Performances of algorithm 2.5.1 are plotted in Graphs 2.7.1-2.7.4. Computation of tissue

parameters using test image set 3 are done following the method proposed by S.B.Rao et al., [60]. First the computations of the tissue parameters are done on the uncompressed images and they are compared with the parameter images computed from compressed images at various levels of compression. Along with the computed images (Image 2.7.4-2.7.7), maps of points where the computations are not carried out are given. The effect of compression in tissue parameter computations is listed in a table (Table 2.7.1). The test images used in these experiments are normalized to take two bytes per pixel. To estimate the amount of distortion induced because of compression decompression, we use two kinds of error measures, they are defined as follows. The amount of compression obtained is measured, by percentage compression (P.C) given by,

$$\text{Percentage Compression (P.C)} = \text{Compression Ratio (C.R)} \times 100$$

Following quantities are used for measuring the error in decompression,

$$\text{Relative l1 Error (rl1)} = \frac{\sum_{\text{all pixels}} |f(i) - \tilde{f}(i)|}{\sum_{\text{all pixels}} |f(i)|} \quad \& \quad \text{Relative l2 Error (rl2)} = \frac{\left( \sum_{\text{all pixels}} |f(i) - \tilde{f}(i)|^2 \right)^{\frac{1}{2}}}{\left( \sum_{\text{all pixels}} |f(i)|^2 \right)^{\frac{1}{2}}}.$$

$$\text{Average l1 Error (al1)} = \frac{\sum_{\text{all pixels}} |f(i) - \tilde{f}(i)|}{\text{Total no. of pixels}}.$$

$$\text{Average l2 Error (al2)} = \frac{\left( \sum_{\text{all pixels}} |f(i) - \tilde{f}(i)|^2 \right)^{\frac{1}{2}}}{\text{Total no. of pixels}}.$$

In the above expressions  $f$  is the uncompressed image and  $\tilde{f}$  is the decompressed image. For the compression of  $T_1$ ,  $T_2$  and spin density weighted images, we store them in a set of three images corresponding to one cross section. The total size of the set is used for calculating the percentage compression.

**55 % Compression using Algorithm 2.5.1**

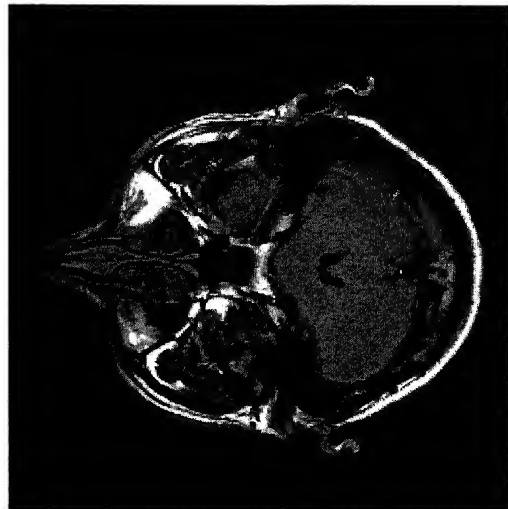
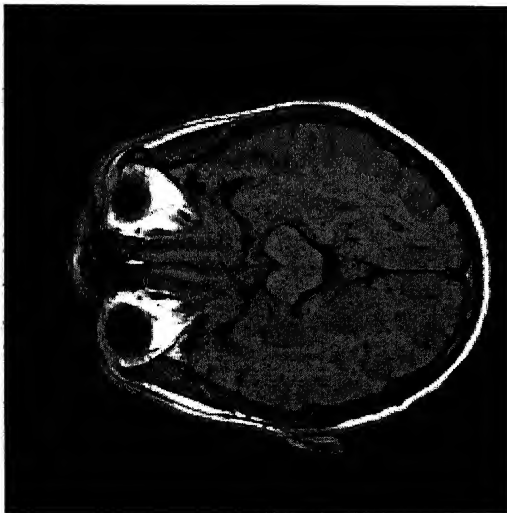
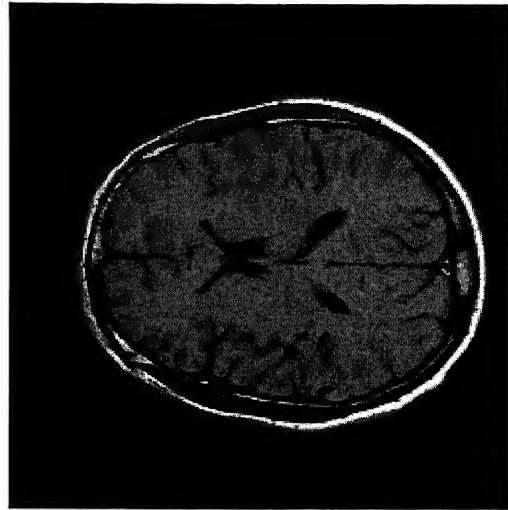
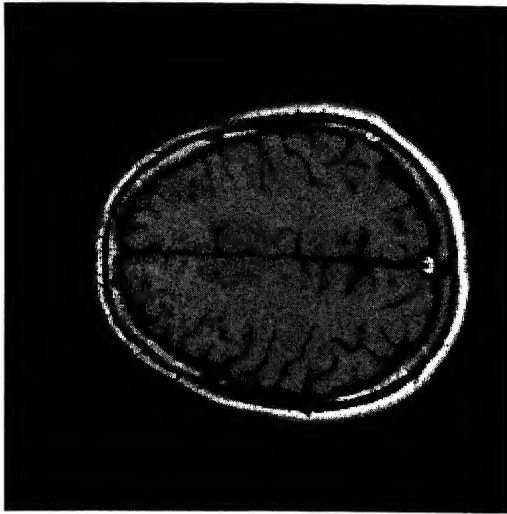


Image 2.7.1



60 % Compression using Algorithm 2.5.1

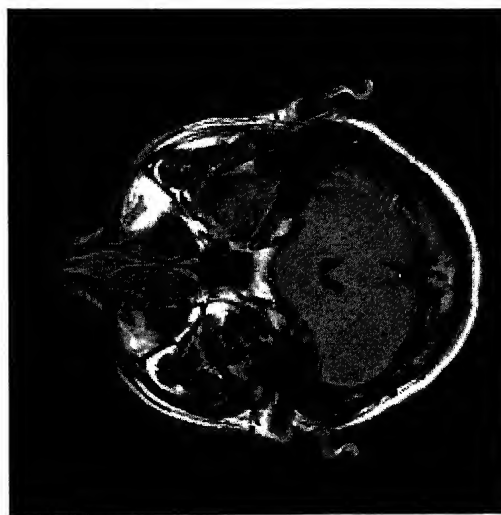
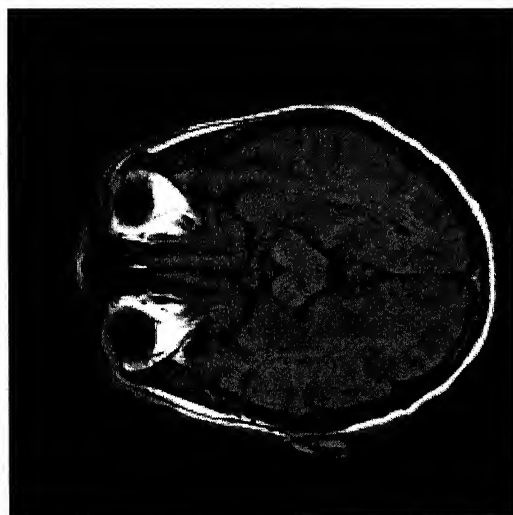
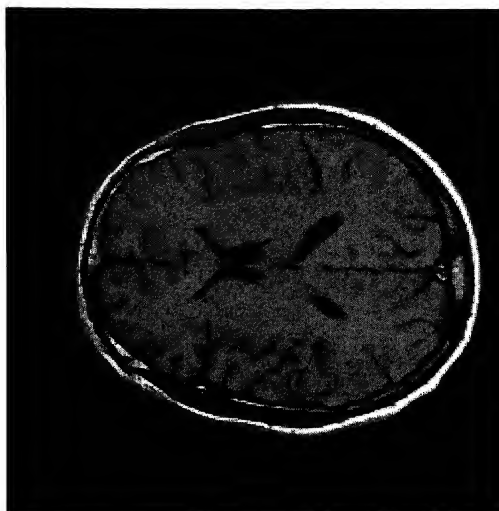


Image 2.7.2

**65 % Compression using Algorithm 2.5.1**

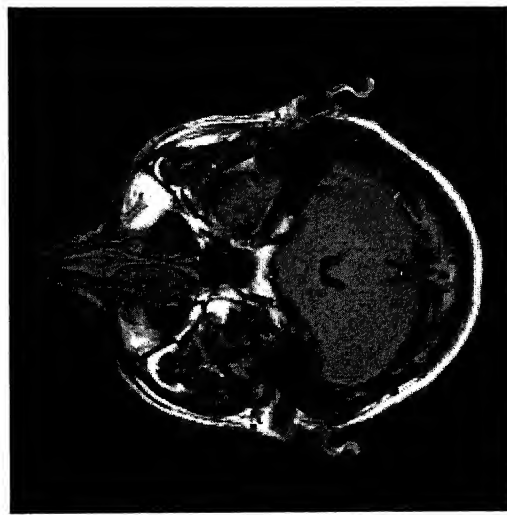
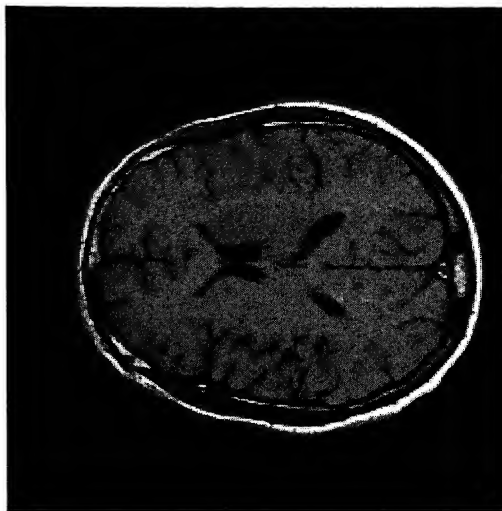
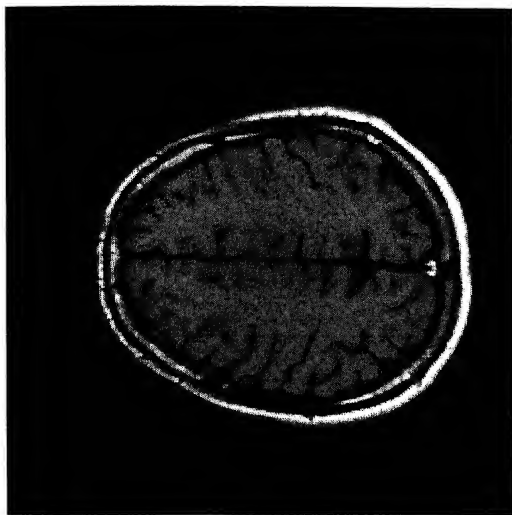
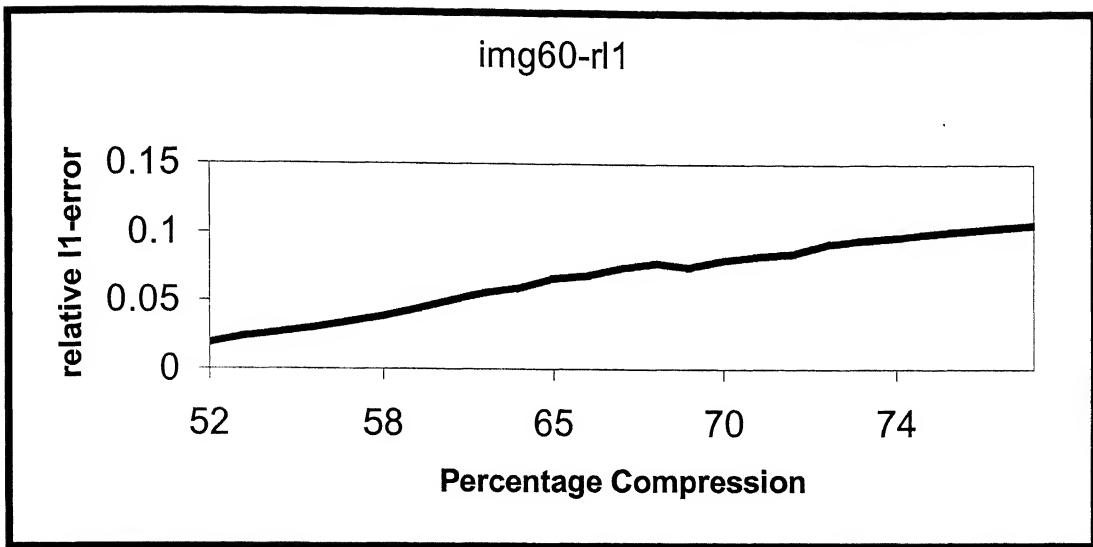
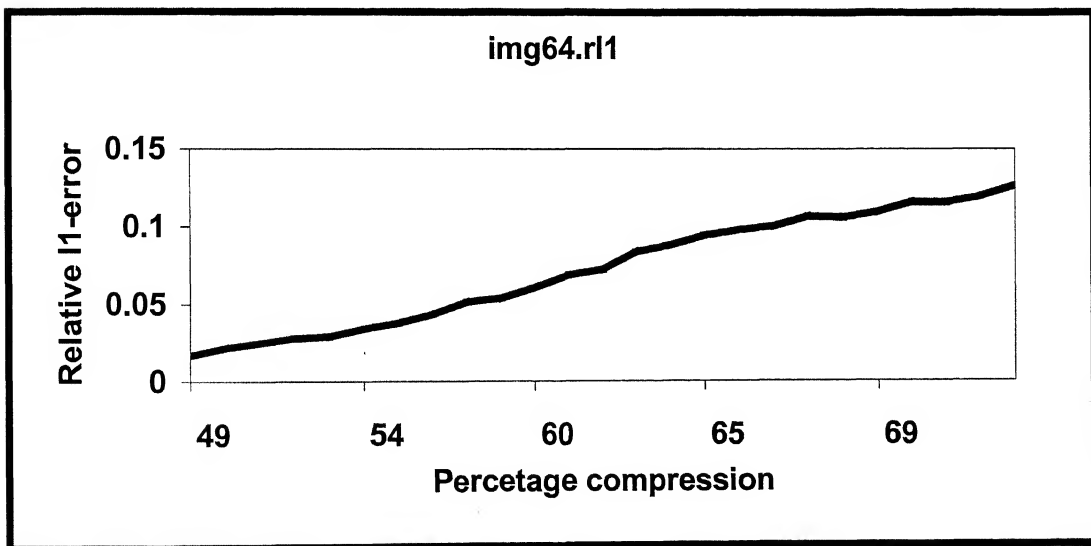


Image 2.7.3

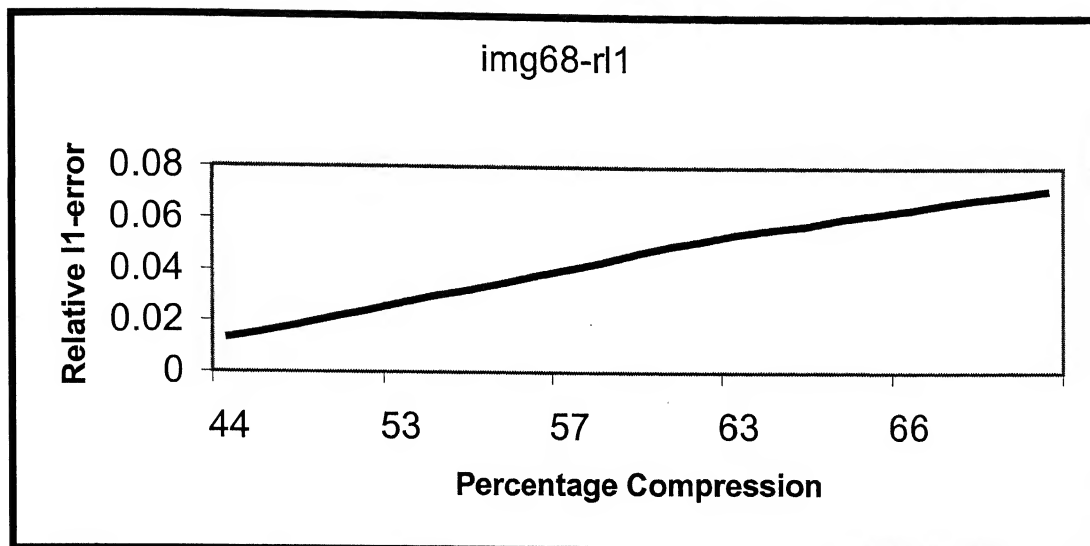
## Percentage Compression Vs Relative Error



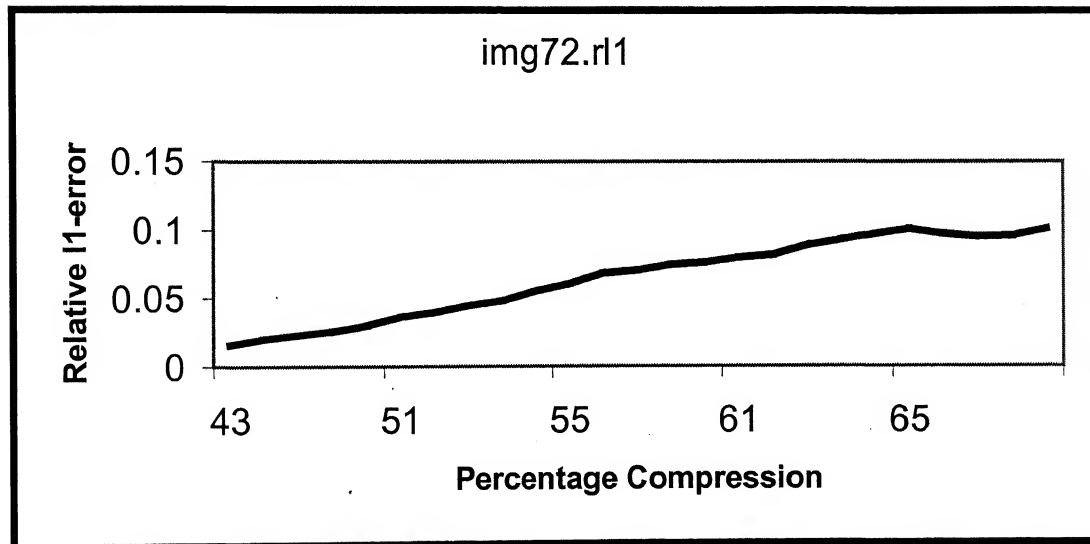
Graph 2.7.1



Graph 2.7.2

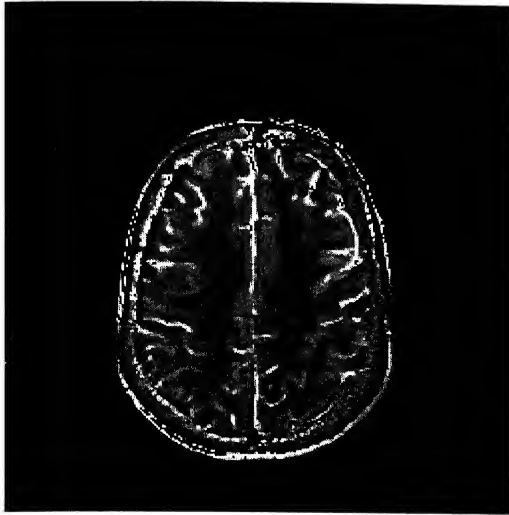


Graph 2.7.3

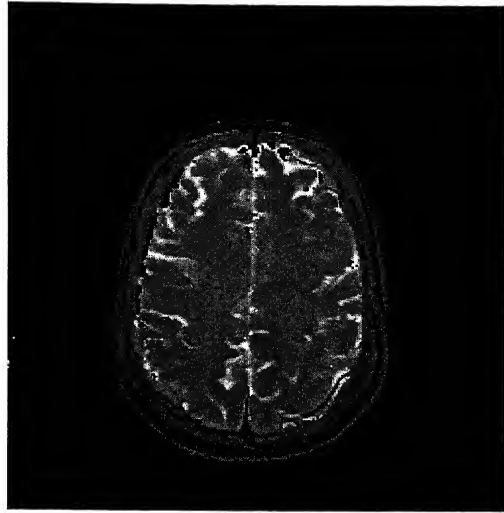


Graph 2.7.4

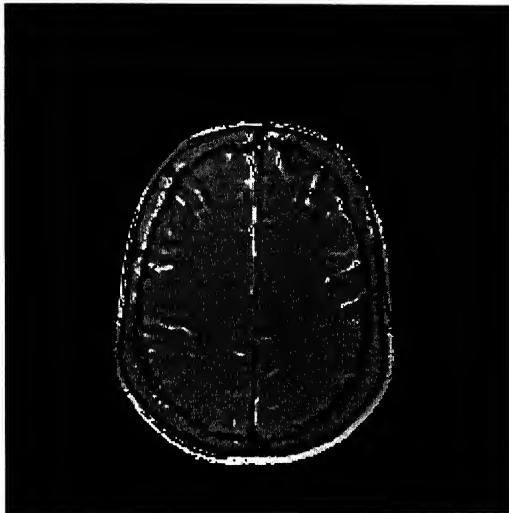
**T1, T2 and Spin Density computed from uncompressed images**



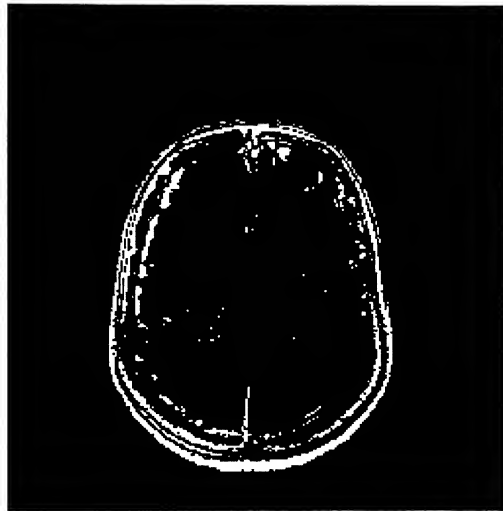
T1



T2



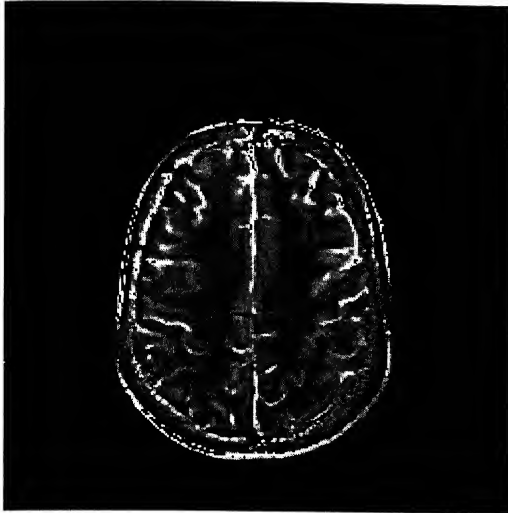
Spin Density



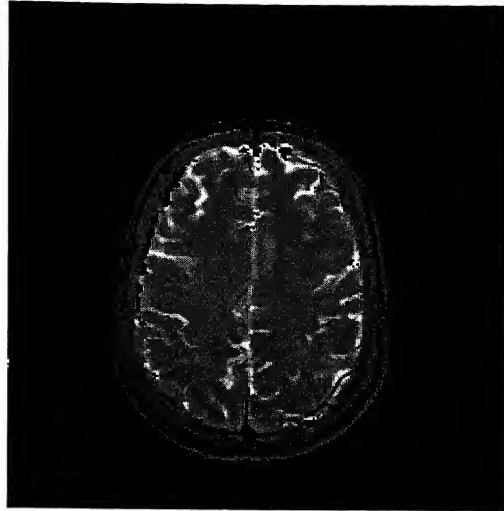
Map of points where computations  
are not carried out

Image 2.7.4

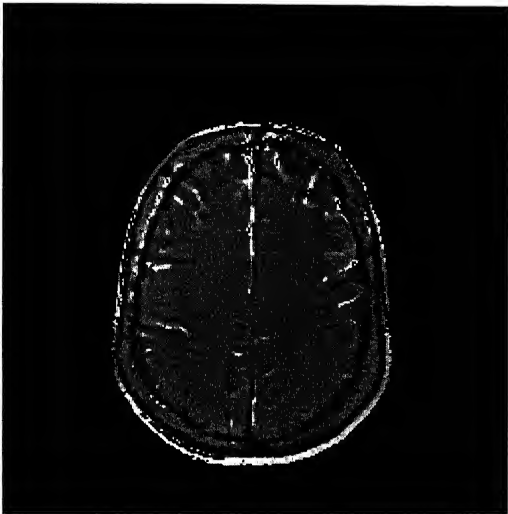
Computed from 47.38 % Compressed image set



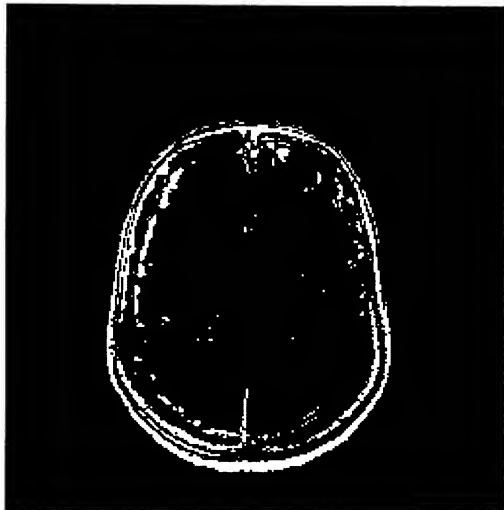
T1



T2



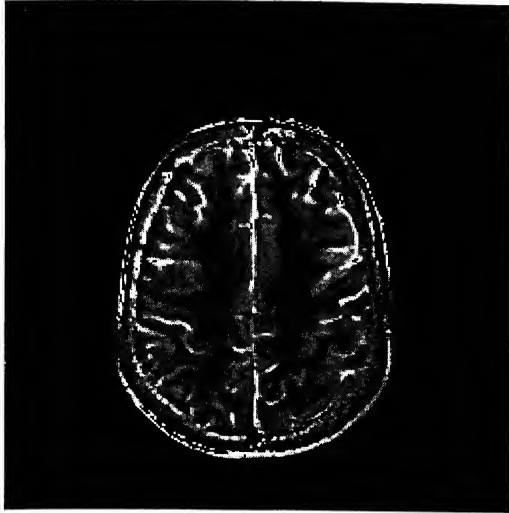
Spin Density



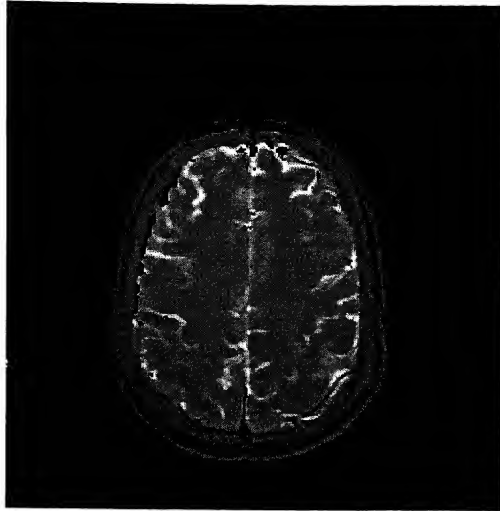
Map of points where computations  
are not carried out

Image 2.7.5

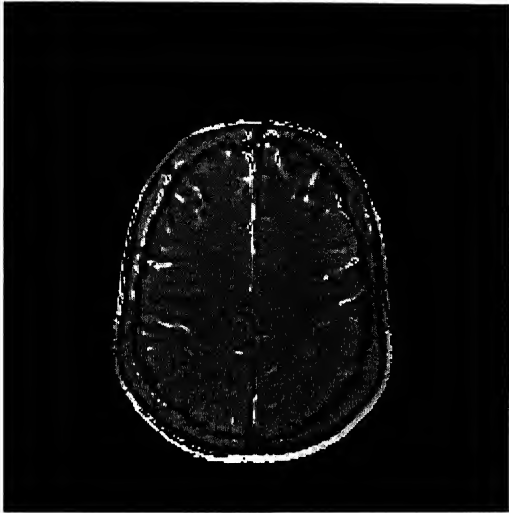
Computed from 57.68 % Compressed Image set



T1



T2



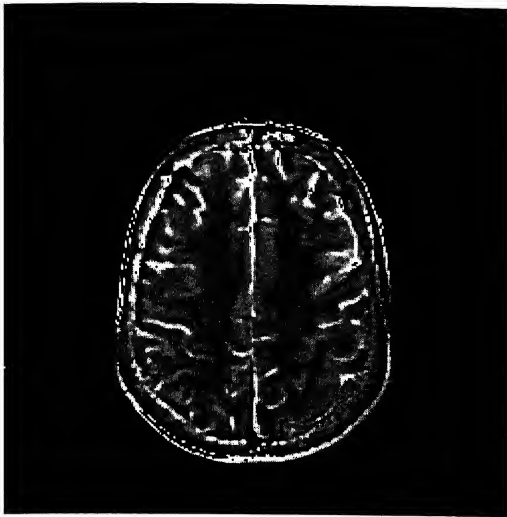
Spin Density



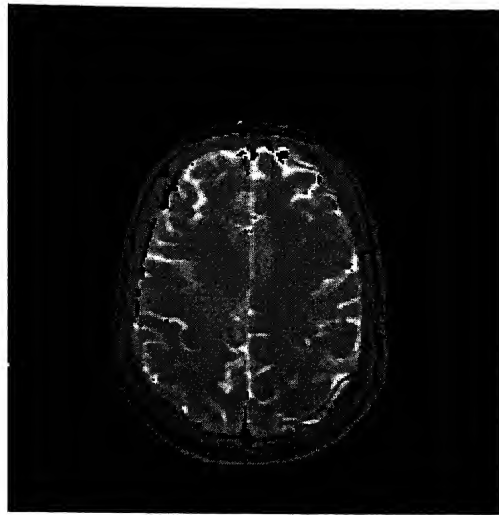
Map of points where computations  
are not carried out

Image 2.7.6

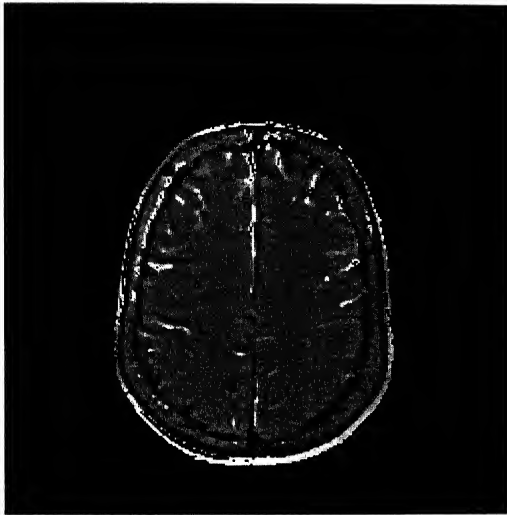
Computed from 66.75 % Compressed image set



T1



T2



Spin Density



Map of points where computations  
are not carried out

Image 2.7.7



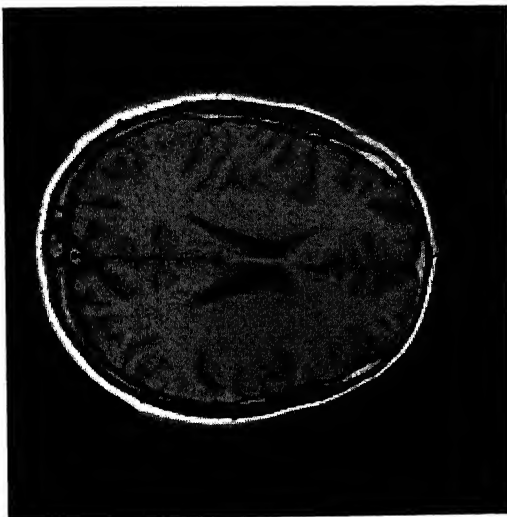
## Effect of compression in the computation of Tissue parameters

Relative errors obtained are with respect to the parameters computed from uncompressed images. In the following table, P.C stands for percentage compression, r11 and r12 stands for relative l1 error and relative l2 error.

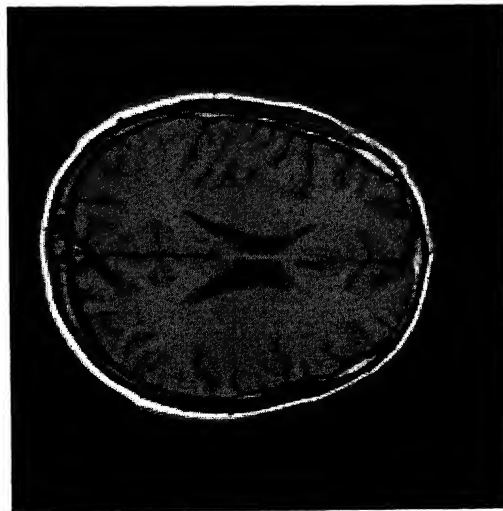
P.C	T1		T2		P.D	
	r11	r12	r11	r12	r11	r12
47.38	0.023001	0.108814	0.038972	0.082052	0.017739	0.110434
57.68	0.058826	0.153803	0.048279	0.117673	0.037774	0.155013
58.60	0.070829	0.173674	0.048764	0.125593	0.040470	0.173440
62.41	0.121785	0.210169	0.061526	0.153636	0.071345	0.205007
66.75	0.151067	0.259694	0.060059	0.158104	0.080254	0.237892
71.20	0.172699	0.309278	0.066849	0.178419	0.093853	0.269737

Table 2.7.1

## Compression of Raw Data

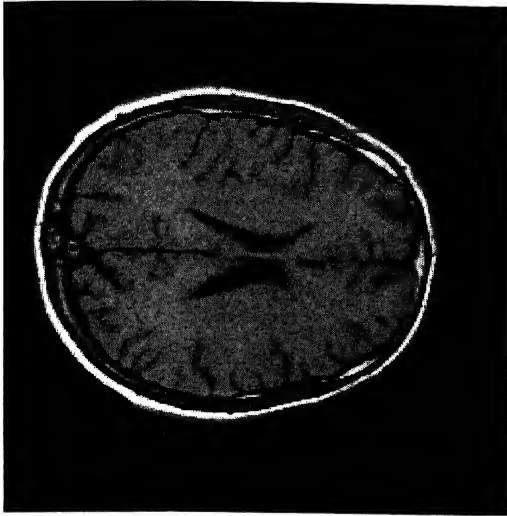


Raw data uncompressed

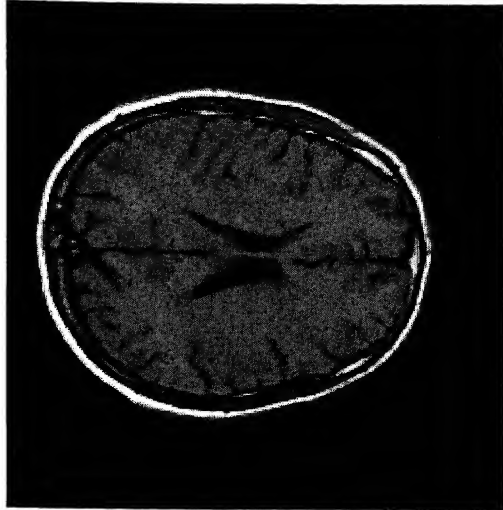


Raw data 65 % compressed

Image 2.7.8



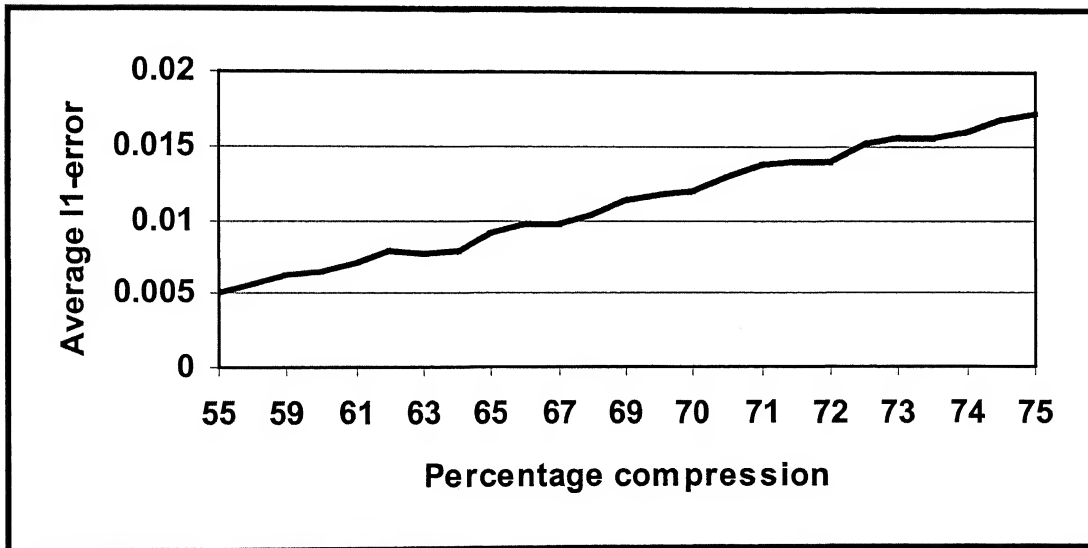
Raw data 70 % compressed



Raw data 75 % compressed

Image 2.7.9

**Percentage compression Vs Average l1-error  
(Compression of Raw data)**



Graph 2.7.5

The decompressed images at 55,60 and 65 % compression levels are shown in Image 2.7.1-2.7.3. The graphs of Percentage compression vs relative error are presented in Graphs 2.7.1-2.7.4. From the images and graphs we observe that there is no visual deterioration of images up to 60 % compression. After that the image architecture deteriorates slightly. Apart from test images performance of algorithm 2.5.1 has been tested on images corresponding to several other cross sections and the results obtained are similar. Upto 60 % compression is suggested for the images under consideration.

In some specific cases of diagnosis, computed  $T_1$ ,  $T_2$  and density images are used. Three images (Test image set 3) which corresponds to the same cross-sections from which these parameters can be calculated in retrospect are compressed and stored as a set. Images 2.7.4-2.7.7 and the corresponding error table shows that upto. 57% compression the 11-relative error after decompression is less than 5 %. At around 67 % compression we note from Table 2.7.1 that the relative 11-error goes upto 15 %.

Images 2.7.8-2.7.9 are reconstructed using compressed raw data. Reconstructed images with compressed raw data are compared with the ones obtained reconstructed from raw data without compression. From the following images and the error graph (Graph 2.7.5) we note that the distortion is not perceivable by human eye up to 65 % compression. Above that level, slight deterioration is observed. The test raw data permits 65 % compression through algorithm 2.5.1 without visual loss of information.

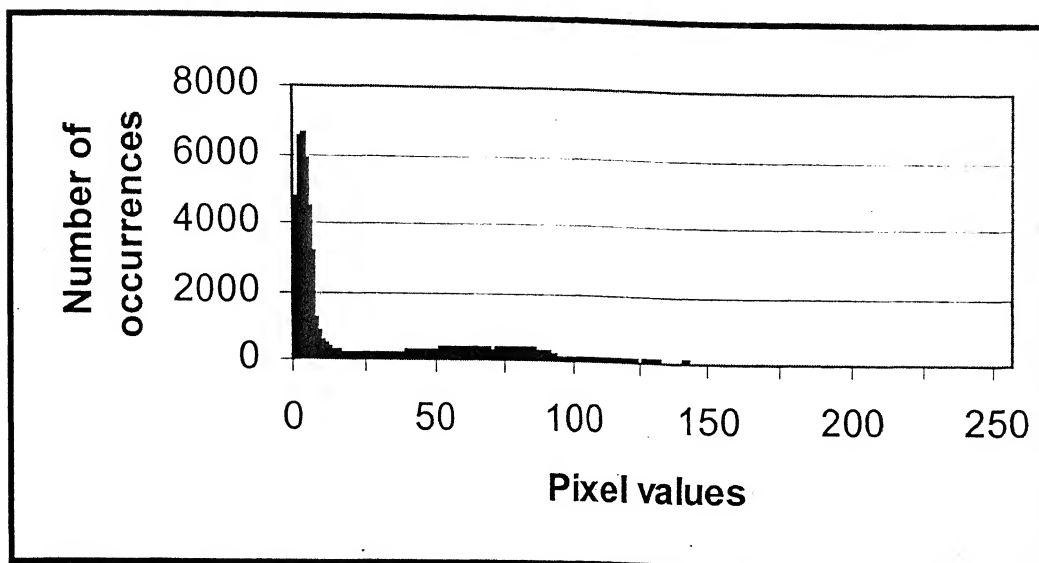
# CHAPTER III: COMPRESSION THROUGH ALGEBRAIC INTERPOLATION

## 3.0 Introduction:

The problem of image compression is viewed as a non-redundant representation of the underlying analog intensity distribution function,  $f$ . A digital image  $\tilde{f}$ , which represents  $f$ , is stored as samples of  $f$  called pixel values. Storage requirements of  $\tilde{f}$  depend on its resolution  $m$ , which is the size of the pixel and depth  $b$ , which is the number of distinct gray levels (states) the image can take. For sufficiently smooth images the number of states taken by the finite differences of  $\tilde{f}$  are less when compared to the number of states taken by the digital image  $\tilde{f}$ . Variable Length Coding (VLC) like Huffman coding [30] uses the peakedness of the histogram by assigning short code word for frequently occurring amplitudes and longer code for others. Consequently, methods based on VLC's are expected to work well for the differences than on the actual images. We illustrate this with the help of an example.

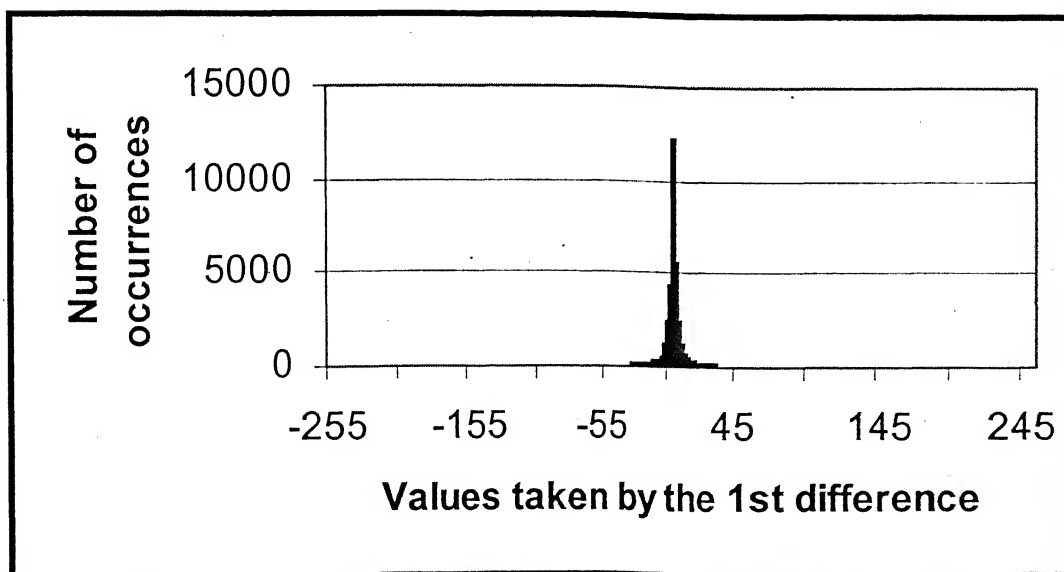
Consider the histogram of the image (Image 1.5.2) and the histogram of its first difference. Graph 3.0.1-3.0.2 (page-73) shows that the number of distinct states required for representing the first difference, is less when compared to that of the image itself. The method of storing first difference instead of pixel values is popularly known as Discrete Pulse Code Modulation (DPCM). It has been investigated since 1970 (Jain, [35]), and the performance of several variations of DPCM has been tested by various researchers (Nijim et al., [54], Ross et al., [63]). In this chapter we propose a compression strategy based on the finite differences for MR images.

### HISTOGRAM OF THE IMAGE



Graph 3.0.1

### HISTOGRAM OF THE FIRST DIFFERENCE



Graph 3.0.2

For smooth images the  $k$ -th difference is sufficiently small and can be coded in lesser number of bits. The smallness of the  $k$ -th difference is used for compression. We quantitatively measure the smoothness of images by their membership in certain smoothness classes  $X^{k,\alpha}$  (defined more precisely in section 2 of this chapter). We choose  $k$ , based on the estimated smoothness of the image. Not much loss of information can be tolerated in medical image compression and these methods render lossless compression. In the present chapter, besides obtaining storage bounds for the proposed compression algorithm we present the error estimates for the representation of  $f$ . Rest of the chapter is organized as follows.

We expect more compression through the finite difference methods, for images, which are sufficiently smooth. In section one, we discuss the method of storing  $k$ -th differences instead of pixel values for compression. Estimation of smoothness for the test images is carried out in section two. This is achieved by measuring the smoothness parameter  $\alpha$  (defined more precisely in the following section). The compression algorithm and storage requirements for the proposed algorithm are presented in the same section. In section three a lemma due to S.N.Bernstein and certain other inequalities, which are useful in obtaining error estimates, are stated and proved. We present the direct error estimates and inverse theorems in section four. In section 5, the experimental implementation of the proposed compression algorithm is carried out on the test images. We show by experiments that for smooth images compression is more.

### 3.1 Storage of Finite Differences:

Throughout this chapter we will use the following notations. Let  $f$  be the digital image stored at a resolution  $m$  and depth  $b$ . Let  $N = 2^m$  and  $h = 2^{-m}$ . The image  $f$  is stored as  $N \times N$  matrix of pixel values, which are some averages of the analog intensity distribution function over small squares of size  $2^{-m} \times 2^{-m}$ . Let  $f(m,n)$  be the  $mn$ -th pixel value.

**Initial Storage:**

$$\begin{array}{cccc}
 f(0,0) & f(0,1) & . & . & . & f(0,N-1) \\
 f(1,0) & f(1,1) & . & . & . & f(1,N-1) \\
 . & . & . & . & . & . \\
 . & . & . & . & . & . \\
 . & . & . & . & . & . \\
 f(N-1,0) & f(N-1,1) & . & . & . & f(N-1,N-1)
 \end{array} \tag{3.1.1}$$

For each fixed  $y$ , the image  $f(x,y)$  is a function of one variable. Throughout this chapter, we consider the image  $f$  as a function of  $y$  for each fixed  $x$ . Storage requirements for an image can be specified in terms of Bit rate, which is defined as follows

**Definition (3.1.1):**

Bit rate is the number of bits required for storing one pixel.

**Bit rate := No. of bits used for storage / No. of pixels used to represent the image.**

Bit rate is given as bits per pixel (bpp). The digital image of depth  $b$  has a bit rate of  $b$ -bpp.

The  $k$ -th horizontal forward difference for the pixel data is given by,

$$\Delta_h^0 f(x, y) := f(x, y)$$

$$\text{and for } j = 1, \dots, k \quad \Delta_h^j f(x, y) := \Delta_h^{j-1} f(x, y + h) - \Delta_h^{j-1} f(x, y). \tag{3.1.2}$$

For a fixed  $k$ , the  $k$ -th horizontal forward differences, are calculated by using the above formula and the differences are stored as follows:

$$\begin{array}{ccccccc}
\Delta_h^0 f(0,0) & \Delta_h^1 f(0,0) & \dots & \Delta_h^k f(0,0) & \Delta_h^k f(0,1) & \dots & \Delta_h^k f(0, N-k-1) \\
\Delta_h^0 f(1,0) & \Delta_h^1 f(1,0) & \dots & \Delta_h^k f(1,0) & \Delta_h^k f(1,1) & \dots & \Delta_h^k f(1, N-k-1) \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\Delta_h^0 f(N-1,0) & \Delta_h^1 f(N-1,0) & \dots & \Delta_h^k f(N-1,0) & \Delta_h^k f(N-1,1) & \dots & \Delta_h^k f(N-1, N-k-1)
\end{array}
\quad (3.1.3)$$

From these differences the actual pixel values for the image can be calculated using,

$$\begin{aligned}
f(r, s) &= \sum_{j=0}^s \binom{s}{j} \Delta_h^j f(r, 0) & 0 \leq s \leq k-1, \quad 0 \leq r \leq N-1 \\
f(r, j) &= \Delta_h^k f(r, j-k) + \sum_{v=1}^k (-1)^{v+1} \binom{k}{v} f(r, j-v) & k \leq j \leq N-1, \quad 0 \leq r \leq N-1
\end{aligned}
\quad (3.1.4)$$

The pixel values calculated by using the above expressions will be exact and there is no loss of information. We theoretically estimate the error between the original image  $f$  and the decompressed digital image. From the stored  $k$ -th differences, pixel values at non-nodal points can be estimated by using a  $k$  point equi-spaced interpolation scheme. These interpolation schemes help us to obtain images on a finer grid and are practically useful. At non-nodal points the following interpolation formula can be used

$$f(x, y) = f(x, 0) + \frac{y}{h} \Delta_h f(x, 0) + \dots + \frac{y(y-h) \dots [y-(k-1)h]}{k! h^k} \Delta_h^k f(x, 0). \quad (3.1.5)$$

Here one argument in  $f(x, y)$  is fixed and  $y_0, \dots, y_k$  are nodal points.

By image compression we are trying to reduce the redundancy in the original representation. For example, the Huffman coding uses the statistical redundancy, i.e. the number of occurrences of each symbol/state and can be calculated from the histogram of the data. The frequency information is already available with the data and this is used for variable length coding of the data. Our objective is to reduce the redundancy due to smoothness, which is a measure of correlation between adjacent pixels. With this objective we first estimate the smoothness, under suitable hypothesis on the images under consideration. Without any apriori knowledge about the smoothness of the images if we



store the  $k$ -th differences instead of the pixel values then it actually would take more space than the original representation. If the original image is coded in  $b$  bits per pixel then the  $k$ -th difference would require  $(b+k)$  bits per pixel. This is stated in the form of a lemma.

**Lemma 3.1.6:** The  $k$ -th horizontal forward difference of a  $b$ -depth image takes  $(b+k)$ -bits at most.

**Proof by induction:**

1<sup>st</sup> difference: If we subtract two  $b$ -bit unsigned numbers (binary) then the result has at most  $b$ -bit magnitude and one sign bit which makes it  $(b+1)$ -bits.

2<sup>nd</sup> difference : If we take the second difference using three  $b$ -bit unsigned numbers, then it introduces at most one carry bit extra and one sign bit. Therefore  $(b+2)$  bits in total.

Let us assume that our claim is true for  $(r-1)$ th difference, the  $r$ -th difference

$$\Delta_h^r f(x, y) := \Delta_h^{r-1} f(x, y + h) - \Delta_h^{r-1} f(x, y)$$

it is the subtraction of two  $r-1$  th difference which already has a sign bit allocated and here it may at most introduce a carry bit extra,  $b+r-1+1=b+r$  bits are required at most.

Although the  $k$ -th difference requires more space than the pixel values, the smoothness of the image allows us to code the  $k$ -th difference in less space than the original pixel values, for suitably chosen  $k$ . This is achieved by estimating the smoothness of images under suitable hypothesis. We fix the value of  $k$  based on the smoothness of the image and code the  $k$ -th difference instead of the pixel values. In the following section we estimate the smoothness of digital images by their membership in suitable smoothness class. The smoothness estimation gives the order of the  $k$ -th differences, this order information is used for compression.

### 3.2 Estimation of Smoothness:

We measure the smoothness of images by their membership in the following class of functions.

**Definition 3.2.1:** For  $0 < \alpha < k$ , the smoothness space  $X^{\alpha,k}$  is the set of functions which have  $\alpha$  smoothness, i.e. the  $k$ -th modulus of smoothness  $\omega_k(\delta)$  is of the order of  $\delta^\alpha$  as  $\delta \rightarrow 0$

$$X^{\alpha,k} = \{f : \omega_k(f, \delta) = O(\delta^\alpha)\}.$$

Let the image  $f \in X^{\alpha,k}$ , the value of  $\alpha$  can be estimated by measuring  $\omega_k(f, \delta)$ , for various values of  $k$ .

$$\omega_k(f, \delta) = O(\delta^\alpha),$$

hence,  $\omega(f, \delta) \leq C \delta^\alpha$ , where  $C$  is a constant. Therefore,

$$\log(\omega_k(f, \delta)) \leq \log C + \alpha \log(\delta). \quad (3.2.1)$$

The value of  $\alpha$  and the associated constant can be estimated by measuring the slope and y-intercept of the graph  $\log(\delta)$  Vs  $\log(\omega_k(f, \delta))$ . For estimating  $\alpha$  we consider  $\omega_k(f, \delta)_1$  and  $\omega_k(f, \delta)_2$  which are defined as follows

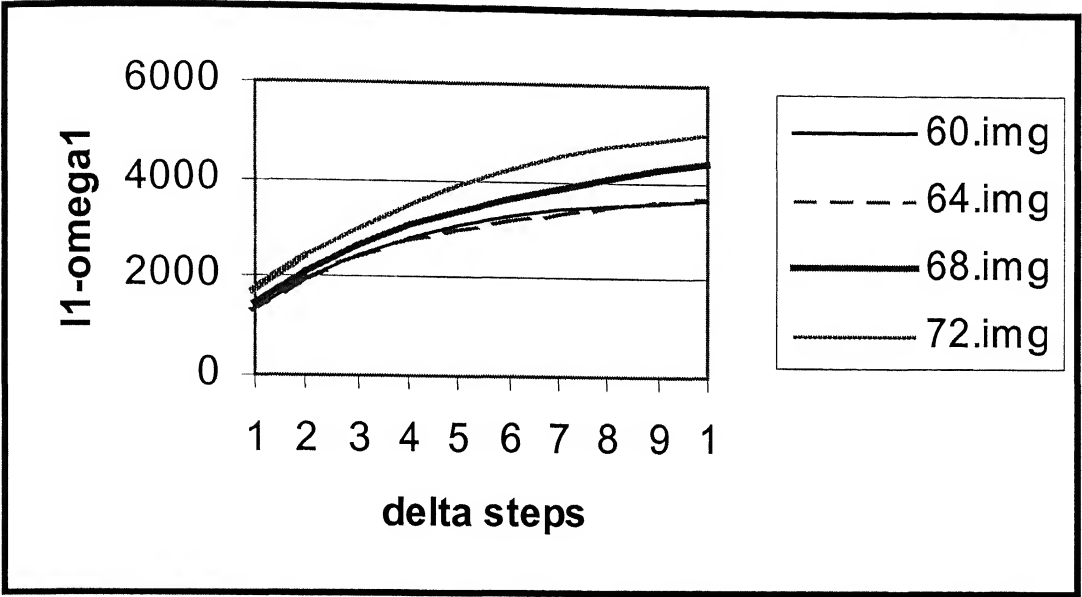
$$\omega_k(f, \delta)_1 = \max_{|t| < \delta} \sum_i |\Delta_i^k f(i)| \quad \text{and}$$

$$\omega_k(f, \delta)_2 = \max_{|t| < \delta} \left( \sum_i |\Delta_i^k f(i)|^2 \right)^{\frac{1}{2}}.$$

For experimental estimation of  $\alpha$  we take  $\delta = rh$ ,  $r = 1, 2, 3, 4, \dots$  i.e. where  $h$  is the resolution of the given image,  $r$  is taken to be  $\delta$  steps. Graphs are drawn between  $\delta$  steps and  $\omega_k(f, \delta)$ . The value of  $\alpha$  and the associated constant  $C$  are estimated by measuring the slope and y-intercept of the graph between  $\log(\delta)$  and  $\log(\omega_k(f, \delta))$ . The smoothness of the test images (test image set 1) are estimated. Throughout this chapter and the remainder of the thesis we consider images of depth 8 bits per pixel, i.e. images with 256 gray levels. Plots of  $\delta$  vs  $\omega_1(f, \delta)_1$  and  $\delta$  vs  $\omega_1(f, \delta)_2$  are shown in Graphs 3.2.1-3.2.2. The plots between  $\log(\delta)$  and  $\log$  of  $\omega_1(f, \delta)_1$ ,  $\omega_2(f, \delta)_1$ ,  $\omega_1(f, \delta)_2$ ,  $\omega_1(f, \delta)_2$  are shown in Graphs 3.2.3-3.2.6. For the test images the log plots are concave and the maximum slope is attained near the origin. The smoothness parameter with respect to  $\omega_k$  for  $k=1, 2$  are the

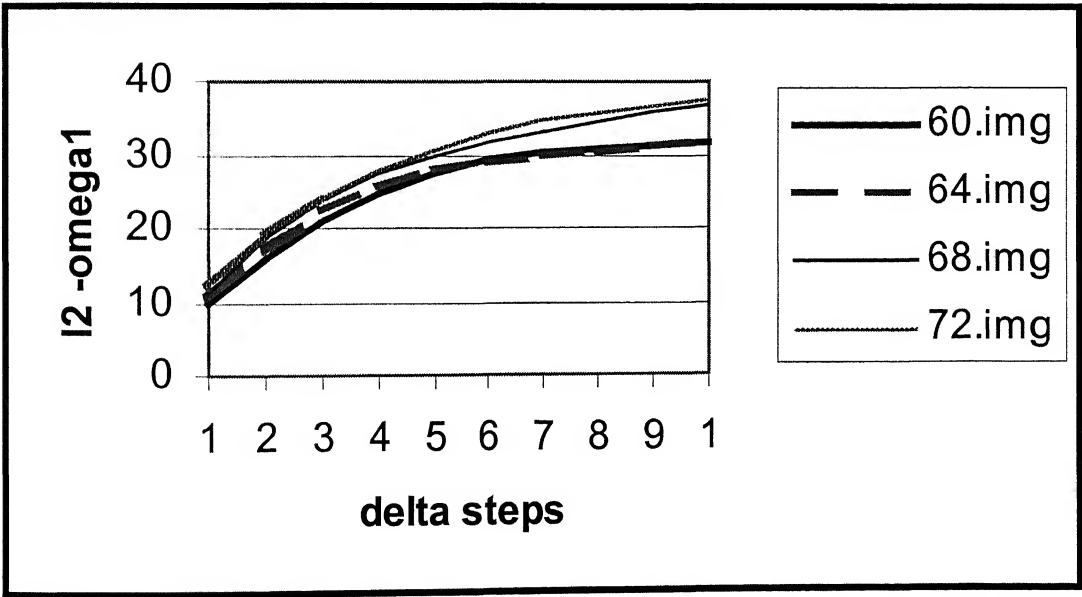
maximum slopes attained in the corresponding log plots. The values of  $\alpha$  and the associated constants are shown in Table 3.2.1-3.2.2.

Graph of  $(\delta)$  Vs  $\omega_1(f,\delta)_1$



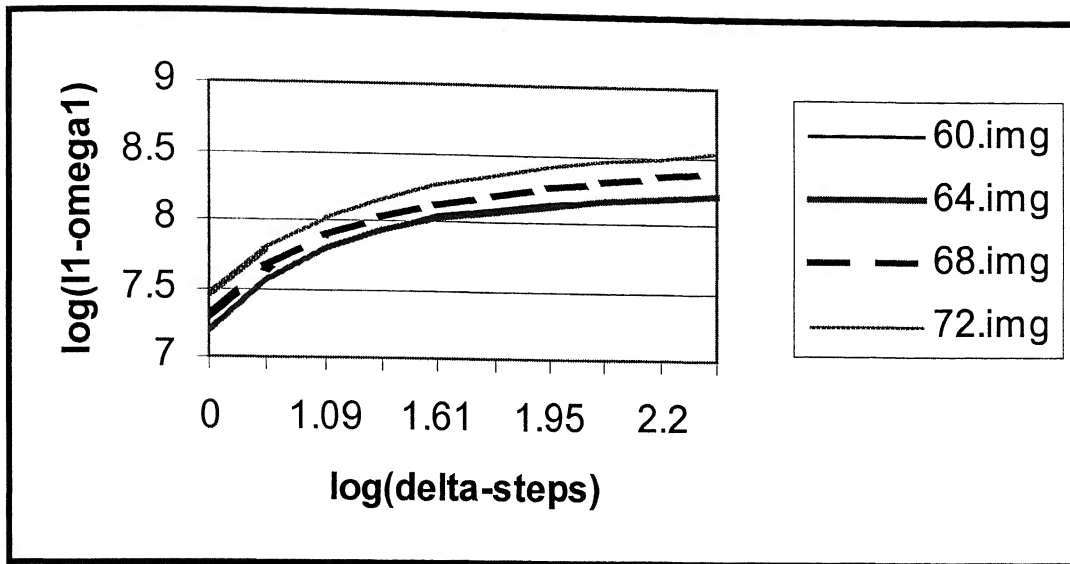
Graph 3.2.1

Graph of  $(\delta)$  Vs  $\omega_1(f,\delta)_2$



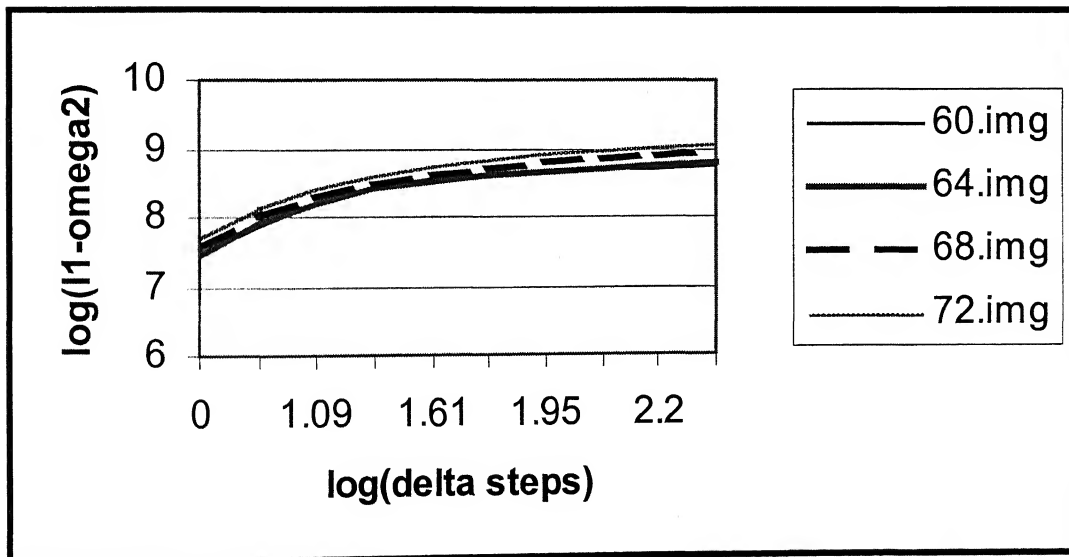
Graph 3.2.2

Graph of  $\log(\delta)$  Vs  $\log(\omega_1(f, \delta)_1)$



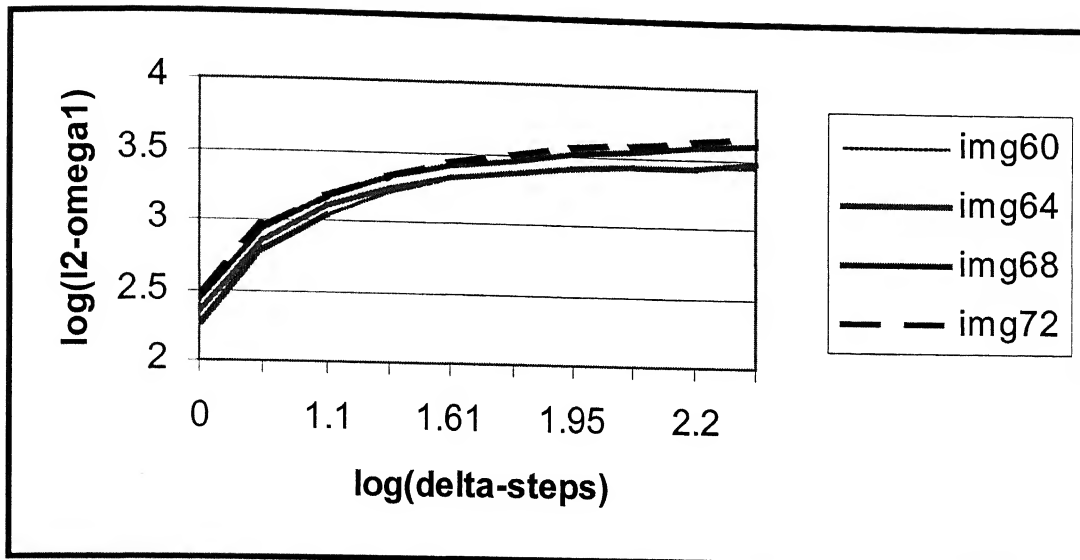
Graph 3.2.3

Graph of  $\log(\delta)$  Vs  $\log(\omega_2(f, \delta)_1)$



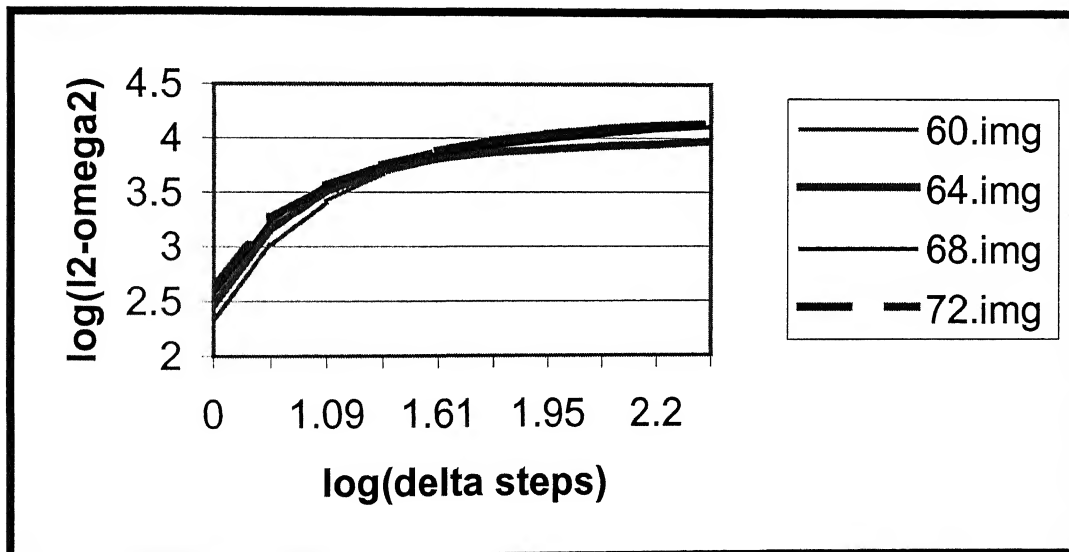
Graph 3.2.4

Graph of  $\log(\delta)$  Vs  $\log(\omega_1(f, \delta)_2)$



Graph 3.2.5

Graph of  $\log(\delta)$  Vs  $\log(\omega_2(f, \delta)_2)$



Graph 3.2.6

### $\alpha$ -Estimates for test image set 1:

Higher value of  $\alpha$  indicates that the image is smoother and lower value of  $\alpha$  indicates the image is less smooth. The estimated values of  $\alpha$  for the test images are presented below. We consider functions which belong to  $X^{k,\alpha}$ , i.e.  $\omega_k(f,h)=O(h^\alpha)$ . Along with the values of  $\alpha$ , the associated constants  $C$ 's are also shown in the table. These constants refer to the constant involved in the  $O$ -notation.

Images	Smoothness measured in l1							
	$\omega_1(f,\delta)_1$		$\omega_2(f,\delta)_1$		$\omega_3(f,\delta)_1$		$\omega_4(f,\delta)_1$	
	$\alpha$	C	$\alpha$	C	$\alpha$	C	$\alpha$	C
60.img	0.544	1322.5	0.586	1709.0	0.529	2902.0	0.490	5292.1
64.img	0.559	1309.8	0.664	1703.5	0.664	2847.6	0.654	5137.7
68.img	0.555	1442.0	0.658	1901.4	0.669	3182.2	0.660	5751.3
72.img	0.556	1653.7	0.646	2178.2	0.648	3658.1	0.651	6610.5

Table 3.2.1

Images	Smoothness measured in l2							
	$\omega_1(f,\delta)_2$		$\omega_2(f,\delta)_2$		$\omega_3(f,\delta)_2$		$\omega_4(f,\delta)_2$	
	$\alpha$	C	$\alpha$	C	$\alpha$	C	$\alpha$	C
60.img	0.743	9.541	0.942	10.45	0.906	16.66	0.855	29.63
64.img	0.727	10.49	1.003	11.77	1.071	18.23	1.087	31.60
68.img	0.717	11.38	0.972	12.96	1.038	20.27	1.051	20.27
72.img	0.666	11.99	0.832	14.59	0.871	23.67	0.889	41.84

Table 3.2.2

The value of  $\alpha$  gives an indication about the number of preceding zeros in the  $k$ -th differences. These preceding zeros need not be stored and consequently the memory space requirements for storing the  $k$ -th difference is less when compared to storing the

actual pixel values. Here the compression is done precisely by not storing the preceding zeros. At the time of decompression, the numbers of preceding zeros are known from the value of  $\alpha$ . Now we present the compression algorithm.

### Algorithm 3.2.1:

**Step 1:** Estimate  $\alpha$  and the associated constant for the given image using the relation (3.2.1).

**Step 2:** Choose  $k$  such that  $0 < b+k+1-m\alpha < b$ .

**Step 3:** Calculate the horizontal forward differences of the image recursively by (3.1.2) for the chosen  $k$ .

**Step 4:** Store differences available in the form (3.1.3) using Huffman coding (cf: 1.2.1).

### Storage requirements:

Let  $f \in X^{k,\alpha}$ . The first  $k$  differences for each row are stored separately. If  $f$  is stored in  $b$  bits per pixel then each  $k$ -th difference  $\Delta_h^k f$  requires  $b+k$  bits for storage. Out of  $b+k$  bits one bit is reserved for storing the sign. We have,

$$|\Delta_h^k f(x)| \leq \omega_k(f, h) = O(h^\alpha).$$

For  $h = 2^{-m}$  the resolution of the image, we have  $|\Delta_h^k f(x)| = O(2^{-m\alpha})$ , therefore the binary representation of the  $k$ -th differences will have  $m\alpha$  preceding zeros which need not be stored, therefore the net bit rate required for storing the  $k$ -th differences is  $b+k-m\alpha$ .

The initial  $(k-1)$  differences occurring in each row of (3.1.3) are stored in the header and the rest of the  $k$ -th differences are coded in  $b+k-m\alpha$  bits. Compression is achieved because  $(b+k+1-m\alpha) < b$ , this follows from the way  $k$  is chosen (as per the algorithm 3.2.1) based on the estimated  $\alpha$ . Consequently the total storage space required for  $N \times N$  size image will be  $(N-k) \times N \times (b+k+1-m\alpha)$ . Compression through algorithm 3.2.1 is a non-lossy. At the nodal points we can exactly get back the original pixel values using equations (3.2.1). The exact implementation of algorithm 3.2.1 is discussed in section 3.5.

### 3.3 Bernstein's Lemma and certain estimates for smooth images:

Here we estimate the error incurred in  $k$ -th difference decomposition by measuring the difference between the original, analog intensity distribution function, and the decompressed digital image. We started with a digital image, which are samples of the original function. After decompression the values at any non-nodal point is estimated using a local  $k$ -th degree Lagrange interpolation polynomial (see Hildebrand [28]),  $L_k$ . In this section we shall prove some lemmas which are used in obtaining error estimates.

**Lemma 3.3.1 (S.N.Bernstein)** (see Timan [71]):

If the continuous function  $f(x)$  on the interval  $[a,b]$  has  $(k+1)$  distinct zeros, then its  $k$ -th difference  $\Delta_{\delta}^k f(x)$  has at least one zero on this segment for sufficiently small  $\delta$ .

Since the original reference of Bernstein was not available to us an inductive proof of the lemma is included below:

Between any two adjacent zeros of  $f$  viz.  $z_1, z_2$  we will show that there exists a  $\xi \in (z_j, z_{j+1})$  such that  $\Delta_{\delta_j}^1 f(\xi) = 0$  for sufficiently small  $\delta_j$ ,

$$f|_{(z_j, z_{j+1})} > 0 \quad (\text{or } < 0)$$

because  $z_j$  and  $z_{j+1}$  are adjacent

$f|_{(z_j, z_{j+1})}$  attains local maxima (or local minima) say  $M$ . Consider lines  $y = r$  ( $0 \leq r \leq M$ ).

For each  $r$ ,  $y = r$  intersects with  $f|_{(z_j, z_{j+1})}$  at least at two points except possibly for the case  $r = M$ .

Let  $p_j$  and  $p_{j+1}$  be two adjacent points of intersection of  $y = r$  and  $f|_{(z_j, z_{j+1})}$ . Without loss of generality let  $p_{j+1} > p_j$  so that  $0 < \delta_j = p_{j+1} - p_j$ . For this  $\delta_j$ ,  $\Delta_{\delta_j}^1 f(p_j) = 0$  and  $\forall \delta < \delta_j$  we have  $\Delta_{\delta_j}^1 f(p) = 0$  for some  $p \in [z_j, z_{j+1}]$ .

$\Rightarrow$  Between any two zeros  $z_j, z_{j+1}$  of  $f$  there exists a zero of  $\Delta_{\delta_j}^1 f(x)$ .



Let  $\delta_1 = \min\{\delta_j\} \ 0 \leq j \leq k$ . For this  $\delta_1 \exists$  at least one point  $\xi_j$  between every pair of adjacent zeros of  $f \ni \Delta_{\delta_1}^1 f(\xi_j) = 0 \ z_j \leq \xi_j \leq z_{j+1}$ .

$\Rightarrow \Delta_{\delta_1}^1 f$  has at least  $k$ -zeros.  $\Delta_{\delta_1}^1 f(x)$  is a continuous function because  $\Delta_{\delta}^1 f(x) = f(x + \delta) - f(x)$  and  $f$  is a continuous function.

$\Rightarrow \Delta_{\delta_1}^1 f$  is a continuous function and has  $k$ -zeros. Therefore by the same above argument  $\exists$  a  $\delta_2' > 0$  such that between any two zeros of  $\Delta_{\delta_1}^1 f$  there exists a zero of  $\Delta_{\delta_2'}^1 (\Delta_{\delta_1}^1 f)$ .

For  $\delta_2 = \min\{\delta_1, \delta_2'\}$ ,  $\Delta_{\delta_2}^1 (\Delta_{\delta_1}^1 f(x)) = \Delta_{\delta_2}^2 f(x)$  has a zero between any two zeros of  $\Delta_{\delta_1}^1 f$  continuing this process we get, there exists a  $\xi \ni \Delta_{\delta}^k f(\xi) = 0$  where  $\delta = \min\{\delta_j\} \ 0 \leq j \leq k$  i.e.  $\delta \leq \min\{|z_j - z_{j+1}|\} = \text{minimum distance between any two adjacent zeros}$

The following lemma gives a measure of the remainder term in interpolation by  $n$ -th degree polynomial in terms of  $(n+1)$ th differences.

### Lemma 3.3.2:

For any real continuous function  $f(x)$ , the following Newton's formula is valid

$$f(x) = f(0) + \frac{x}{h} \Delta_h f(0) + \dots + \frac{x(x-h) \dots [x-(n-1)h]}{n! h^n} \Delta_h^n f(0) + R_{n+1}(x),$$

$$\text{where } R_{n+1}(x) = \frac{x(x-h) \dots (x-nh)}{(n+1)! \delta^{n+1}} \Delta_{\delta}^{n+1} f(\xi), \quad \text{with } 0 < \delta < h.$$

### Proof :

$$\text{Let, } P(x) = f(0) + \frac{x}{h} \Delta_h f(0) + \dots + \frac{x(x-h) \dots [x-(n-1)h]}{n! h^n} \Delta_h^n f(0)$$

$$\text{Consider } g(t) = f(t) - P(t) - [f(x) - P(x)] \frac{(t-x_0) \dots (t-x_n)}{(x-x_0) \dots (x-x_n)} \quad (3.3.1)$$

Equation (3.3.1) is continuous and has  $(n+2)$  zeros viz.,  $t = x_0 \dots x_n, x$ , where the points  $x_0 \dots x_n$  are equi spaced (spacing  $h$ ) and  $x$  is at most  $h$  distance away from one of the  $x_j$ 's. Therefore the maximum distance between any two adjacent zeros of  $g(t)$  is  $h$ . By Lemma

3.3.1, for the function  $g$ ,  $\exists$  a  $\xi \ni \Delta_\delta^{n+1}g(\xi) = 0$  and  $0 < \delta < h$ . Applying  $\Delta_\delta^{n+1}$  to (3.3.1) at the point  $\xi$  we get

$$0 = \Delta_\delta^{n+1}g(\xi) = \Delta_\delta^{n+1}f(\xi) - [f(x) - P(x)] \frac{(n+1)!\delta^{n+1}}{(x-x_0)\cdots(x-x_n)}, \text{ hence}$$

$$f(x) - P(x) = \frac{(x-x_0)\cdots(x-x_n)}{(n+1)!\delta^{n+1}} \Delta_\delta^{n+1}f(\xi)$$

$$= \frac{x(x-h)\cdots[x-(n-1)h]}{(n+1)!\delta^{n+1}} \Delta_\delta^{n+1}f(\xi) \quad [0 < \delta < h].$$

Now we present the interpolation process, which is adopted for evaluating the pixel values at non-nodal points.

### Process:

Here we process each row of the image separately. The function under consideration is that of one variable, keeping the other one fixed and here interpolation refers to construction of interpolation polynomial using equi distant data. Let  $f \in C_{2\pi}$  and be approximated by  $k$ -th degree algebraic interpolation polynomial  $L_k^f$ . At finitely many equi spaced points the value of  $f$  is known. We partition  $I=[0,2\pi]$  into  $I_1, \dots, I_n$  such that each  $I_j$  contains  $(k+1)$  nodes. In each  $I_j$ ,  $f$  is approximated by  $L_{k,j}^f$ , the  $k$ -th degree interpolation polynomial constructed using the  $(k+1)$  equi spaced nodes.

$$L_{k,h}^f(x) = \sum_{i=1}^n X_{I_i}(x) L_{k,h,j}^f(x) \quad (3.3.2)$$

$$L_{k,h,j}^f(x) = \sum_{i=0}^k l_{j+i}(x) f(x_{j+i})$$

$$l_{j+i}(x) = \frac{(x-x_{j+0})\cdots(x-x_{j+i-1})(x-x_{j+i+1})\cdots(x-x_{j+k})}{(x_{j+i}-x_{j+0})\cdots(x_{j+i}-x_{j+i-1})(x_{j+i}-x_{j+i+1})\cdots(x_{j+i}-x_{j+k})} \quad (3.3.3)$$

$L_{k,2^{-m}}^f$  is the interpolating polynomial on the whole of  $f$ .  $X_{I_j}$  is the characteristic function on  $I_j$ . This amounts to piece wise approximation of  $f$  by interpolation.

**Theorem 3.3.1:**

If  $L_{k,h}^f$  is the decompression of  $f \in [x_0 - h, x_k + h]$ , whose digital image at resolution  $m$  ( $h = 2^{-m}$ ) is used for compression then there exists a  $\delta$ ,  $0 < \delta < h$  such that

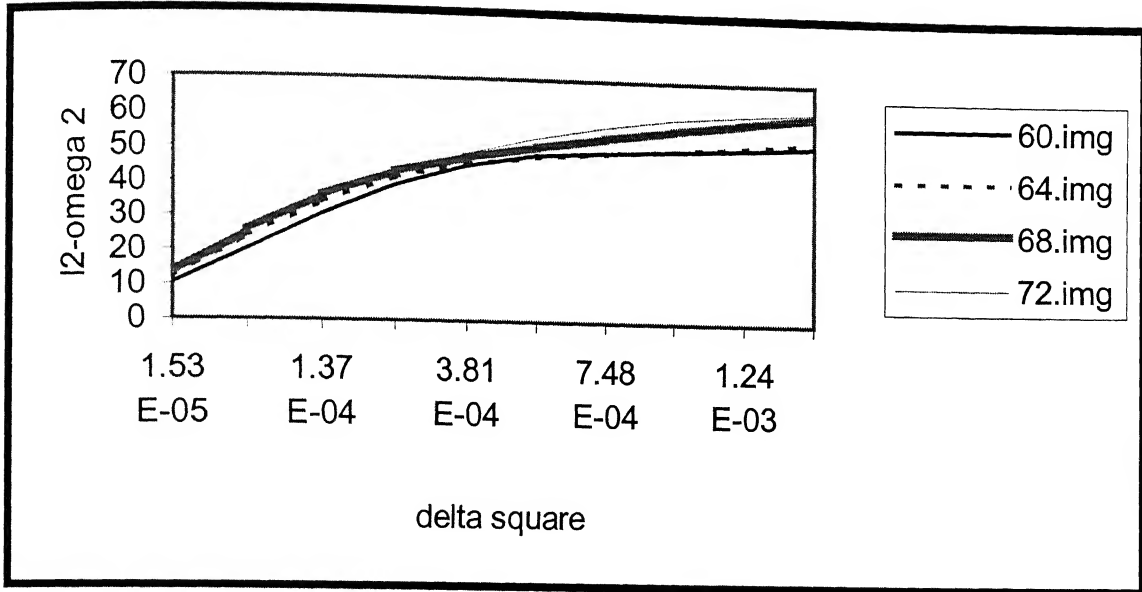
$$|f(x) - L_{k,h}^f(x)| \leq h^{k+1} \frac{\omega_{k+1}(f, \delta)}{\delta^{k+1}}. \quad (3.3.4)$$

**Proof:** Follows from Lemma (3.3.2).

**Remark:**

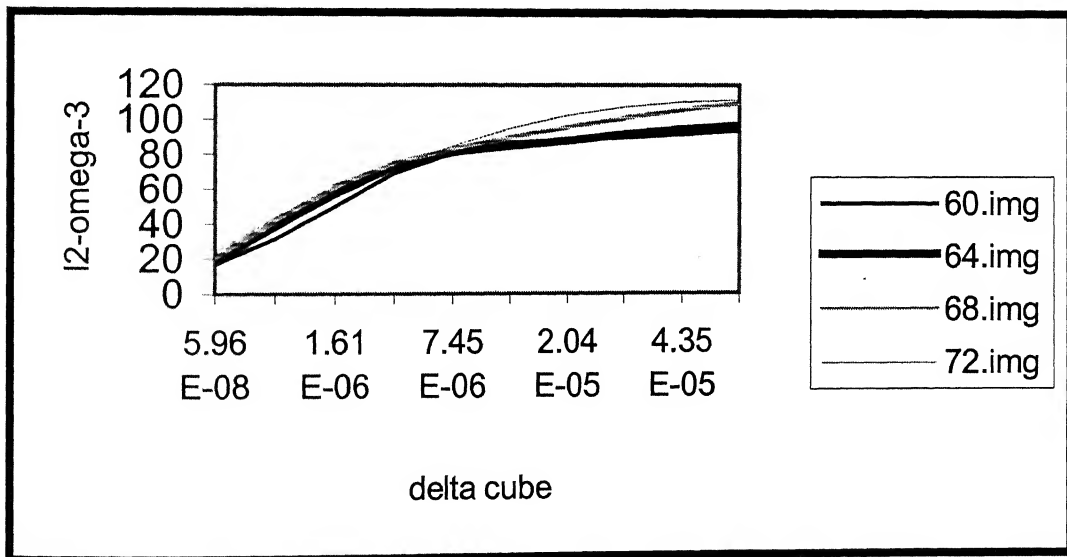
The right hand side in the expression (3.3.4) has  $h^{k+1} \frac{\omega_{k+1}(f, \delta)}{\delta^{k+1}}$ . This could be large for general functions, however for the images under consideration (test image set I) we see from the following graphs 3.3.1-3.3.2, that the slope is maximum near the origin and the shape of the graph is concave. Therefore for the test images under consideration a bound based on the slopes could be obtained from graph of  $\omega_{k+1}(f, \delta)$  vs  $\delta^{k+1}$ . For  $k=1,2$  the value of right hand side comes out to be approximately 0.03. Theorem 3.3.1 says that for  $k = 1,2$  the error incurred in the process is around 3 % for the test images under consideration. In the next section, we obtain the precise order of error in our compression scheme.

Graph of  $\delta^2$  vs  $\omega_2(f;\delta)$  for test images



Graph 3.3.1

Graph of  $\delta^3$  vs  $\omega_3(f;\delta)$  for test images



Graph 3.3.2

**Lemma 3.3.3:**

If  $f$  is a bounded function in  $[x_0 - h, x_k + h]$  then the  $k$ -th degree Lagrange interpolations,  $L_k$ , are uniformly bounded operators, independently of  $h$ .

**Prof:**

$$L_k^f(x) = \sum_{i=0}^k l_i(x) f(x_i),$$

$$\text{where } l_i(x) = \frac{(x - x_0) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_k)}{(x_i - x_0) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_k)}.$$

$$\text{Hence, } |l_i(x)| \leq \frac{C_1 h^k}{C_2 h^k} \leq C_3$$

$$\|L_k^f\|_{\infty} \leq C_k \|f\|_{\infty}$$

where  $C_k$  is a constant independent of  $h$  and depends only on  $k$ . The result follows from this.

**3.4 Direct and Inverse Theorems:**

In this section we derive direct and inverse theorems for the process presented in the previous section. The direct theorem is derived using Peetre's  $K$ -functional, its equivalence with the modulus of smoothness, and certain properties of modulus of smoothness whose short proof for the  $L_p$ -version, adapted from the sup norm case in Lorentz [46], are included for the sake of completeness.

**Definition (3.4.1) (Peetre's  $K$ -functional):** For  $1 \leq p \leq \infty$ , the Peetre's  $K$ -functional is the quantity

$$K(f, t)_p = \inf_{g \in C^{k+1}} \left\{ \|f - g\|_p + t^{k+1} \|g^{(k+1)}\|_p \right\}.$$

This is equivalent to the usual  $(k+1)$ -th modulus of smoothness  $\omega_{k+1}(f, t)_p$ . This is stated and proved as a theorem.

**Theorem 3.4.1:** For  $1 \leq p \leq \infty$ , the Peetre's  $K$ -functional is equivalent to the modulus of smoothness,  $K(f, t)_p \sim \omega_{k+1}(f, t)_p$  i.e.,  $\exists$  constants  $C_1, C_2 > 0$  and are independent of  $t$  such that

$$C_1 \omega_{k+1}(f, t)_p \leq K(f, t)_p \leq C_2 \omega_{k+1}(f, t)_p.$$

To prove this we need the following lemmas,

**Lemma 3.4.1:** For  $1 \leq p \leq \infty$ , the inequality  $\omega_k(f, t)_p \leq 2^k \|f\|_p$  is valid.

**Proof:** For  $1 \leq p < \infty$ ,

$$\begin{aligned} \omega_k(f, t)_p &= \sup_{|h| < t} \left( \int_0^{2\pi} |\Delta_h^k f(x)|^p dx \right)^{\frac{1}{p}} = \sup_{|h| < t} \left( \int_0^{2\pi} \left| \sum_{r=0}^k (-1)^{k-r} \binom{k}{r} f(x+rh) \right|^p dx \right)^{\frac{1}{p}} \\ &\leq \sum_{r=0}^k \binom{k}{r} \left( \int_0^{2\pi} |f(x+rh)|^p dx \right)^{\frac{1}{p}} \leq 2^k \|f\|_p. \end{aligned}$$

For  $p = \infty$ , we refer Lorentz [46].

**Lemma 3.4.2:** For  $1 \leq p \leq \infty$ ,  $f \in C^k$  and  $f^{(k)} \in L_p[0, 2\pi]$  the inequality  $\omega_k(f, t)_p \leq t^k \|f^{(k)}\|_p$  is valid.

**Proof:** For  $1 \leq p < \infty$ ,

$$\begin{aligned} \Delta_t^k f(x) &= \int_0^t \cdots \int_0^t f^{(k)}(x + y_1 + \cdots + y_k) dy_1 \cdots dy_k \\ \left( \int_0^{2\pi} |\Delta_t^k f(x)|^p dx \right)^{\frac{1}{p}} &= \left( \int_0^{2\pi} \left| \int_0^t \cdots \int_0^t f^{(k)}(x + y_1 + \cdots + y_k) dy_1 \cdots dy_k \right|^p dx \right)^{\frac{1}{p}} \\ &\leq \int_0^t \cdots \int_0^t \left( \int_0^{2\pi} |f^{(k)}(x + y_1 + \cdots + y_k)|^p dx \right)^{\frac{1}{p}} dy_1 \cdots dy_k \quad (\text{By Minkowski's ineq.}) \\ &\leq t^k \|f^{(k)}\|_p. \end{aligned}$$

We now present the proof of Theorem 3.4.1.

**Proof of Theorem 3.4.1:**

Let  $f_{t,k+1}$  be the  $(k+1)$ -th Steklov mean (its definition and properties are presented in chapter 2).

$$\begin{aligned}
 K(f, t)_p &\leq \|f - f_{t,k+1}\|_p + t^{k+1} \|f_{t,k+1}^{(k+1)}\|_p \\
 &\leq \omega_{k+1}(f, t)_p + t^{k+1} C_1 t^{-(k+1)} \omega_{k+1}(f, t)_p \\
 &\leq C_2 \omega_{k+1}(f, t)_p
 \end{aligned}
 \quad [\text{Where } C_2 = 2 \max \{1, C_1\}]$$

Conversely, we choose  $g \in C^{k+1}$

$$\begin{aligned}
 \omega_{k+1}(f, t)_p &\leq \omega_{k+1}(f - g, t)_p + \omega_{k+1}(g, t)_p \\
 &\leq 2^{k+1} \|f - g\|_p + t^{k+1} \|g^{(k+1)}\|_p \quad [\text{By lemma 3.4.1 and 3.4.2}] \\
 &\leq 2^{k+1} \left\{ \|f - g\|_p + t^{k+1} \|g^{(k+1)}\|_p \right\} \leq 2^{k+1} K(f, t)_p.
 \end{aligned}$$

**Remark:**

We prove the direct and inverse theorem for a general function for the case  $p = \infty$ . For the case  $1 \leq p < \infty$ , the Lagrange's interpolation operator is not a bounded operator over  $C_{2\pi}$ . However, an MRI image corresponds to a trigonometric polynomial in two variables. Our present compression implementation corresponds to the rows of the image matrix giving rise to trigonometric polynomials in one variable, namely  $x$ , of a fixed order. Since the space,  $\mathfrak{T}_n$  of trigonometric polynomials of order  $n$  is finite dimensional for the MRI purposes, the point functionals  $T_n(x_k)$  are  $L_p$ -bounded:

$$|T_n(x_k)| \leq |T_n * D_n(x_k)|$$

[  $D_n$  stands is the Dirchlet kernel and  $*$  stands for convolution.]

$$\leq \|T_n\|_p \|D_n\|_{p'}, \text{ with } 1 \leq p < \infty \text{ and } \frac{1}{p} + \frac{1}{p'} = 1 \quad [\text{By Holders inequaltiy}]$$

**Lemma 3.4.3:**

Let  $f$  be a MRI image, then for  $1 \leq p \leq \infty$ , the  $k$ -th degree Lagrange interpolations,  $L_k$ , are  $L_p$ -bounded operators, independently of  $h$ .

**Proof:**

By the remark above this lemma, we see that the point functionals for the MRI purposes are  $L_p$ -bounded. The proof of the above lemma follows on the same lines of lemma 3.3.3.

**Lemma 3.4.4:**

Let  $g \in C^{k+1}[x_0 - h, x_k + h]$ . If  $L_{k,h}^g$  be the  $k$ -th degree Lagrange interpolation polynomial of  $g$ , then for  $1 \leq p \leq \infty$ , the following inequality holds:

$$\|g - L_{k,h}^g\|_p \leq C_k h^{k+1} \|g^{(k+1)}\|_p, \text{ where } C_k \text{ is a constant depending only on } k.$$

**Proof:**

The result for the case  $p = \infty$ , is well known (see Hildebrand [28]). We shall prove the result for  $p = 1$ . From these two cases, the result for all  $p$ , ( $1 < p < \infty$ ) follows by the application of the Riesz-Thorin interpolation theorem (see Bergh and Lofstorm [5]).

Let  $P_k$  be the  $k$ -th degree Taylor's polynomial of  $g$ , about a point  $a \in [x_0 - h, x_k + h]$ , say.

Then we have,

$$\|g - L_{k,h}^g\|_1 = \|g - P_k + P_k - L_{k,h}^g\|_1 \leq \|g - P_k\|_1 + \|L_{k,h}^{g-P_k}\|_1 \leq (1 + \|L_k\|) \|g - P_k\|_1$$

Using the integral form of remainder in the Taylor's theorem, we have

$$\begin{aligned} \|g - P_k\|_1 &= \int_{x_0-h}^{x_k+h} |g(x) - P_k(x)| dx \leq \frac{1}{k!} \int_{x_0-h}^{x_k+h} \int_{x_0-h}^x |(x-s)|^k |g^{(k+1)}(s)| ds dx \\ &\leq \frac{1}{k!} \int_{x_0-h}^{x_k+h} |g^{(k+1)}(s)| \int_{x_0-h}^{x_k+h} |(x-s)|^k dx ds \\ &\leq \frac{1}{k!} ((k+2)h)^{k+1} \|g^{(k+1)}\|_1 \\ &\leq C_k h^{k+1} \|g^{(k+1)}\|_1 \end{aligned}$$

The constant  $C_k$  depends only on  $k$ .



### Theorem 3.4.2 (Direct):

If  $L_{k,h}^f$  is the decomposition of  $f \in C_{2\pi}$ , at any arbitrary point of  $f$  whose digital MR image at resolution  $m$ , i.e.  $h=2^{-m}$  is used for compression then the following estimate holds

$$\|f - L_{k,h}^f\|_p \leq C \omega_{k+1}(f, h)_p,$$

where the constant  $C$  is independent of  $f$  and is dependent only on  $k$ .

**Proof:**

We consider a function  $g \in C_{2\pi}^{k+1}$  with  $g^{(k+1)}$  continuous and  $\|g^{(k+1)}\|_p$  is finite, we have

$$\begin{aligned} \|f - L_{k,h}^f\|_p &= \|f - g + g - L_{k,h}^g + L_{k,h}^g - L_{k,h}^f\|_p \\ &\leq \|f - g\|_p + \|g - L_{k,h}^g\|_p + \|L_{k,h}^g - L_{k,h}^f\|_p \\ &\leq (1 + \|L_{k,h}\|) \|f - g\|_p + C_k h^{k+1} \|g^{(k+1)}\|_p \quad [\text{By lemma 3.4.4}] \\ &\leq C_1 K(f, h)_p \\ &\leq C_k \omega_{k+1}(f, h)_p \quad [\text{By theorem 3.4.1}] \end{aligned}$$

The constant  $C_k$  depends only on  $k$ .

### Theorem 3.4.3 (Inverse):

For  $1 \leq p \leq \infty$ , the following inequality holds

$$\omega_{k+1}(f, h)_p \leq C \|f - L_{k,h}^f\|_p,$$

where the constant  $C$ , is independent of  $h$  and  $f$ .

**Proof:**

For a given  $h > 0$ , we construct Lagrange's interpolation polynomial  $L_{k,h}^f$  from the available samples of  $f$ .  $L_{k,h}^f$  is an algebraic polynomial of degree  $k$ , therefore

$$\omega_{k+1}(f, h)_p = \omega_{k+1}(f - L_{k,h}^f, h)_p \leq 2^{k+1} \|f - L_{k,h}^f\|_p.$$

Combining the results of Theorems 3.4.2-3.4.3, it follows that the error in our interpolation scheme is precisely of the order of  $\omega_{k+1}(f, h)_p$ .

### 3.5 Experimental Results:

Algorithm 3.2.1 is implemented on the test images and here we present the results after loss less compression. Algorithm 3.2.1 is implemented in integer arithmetic as follows:

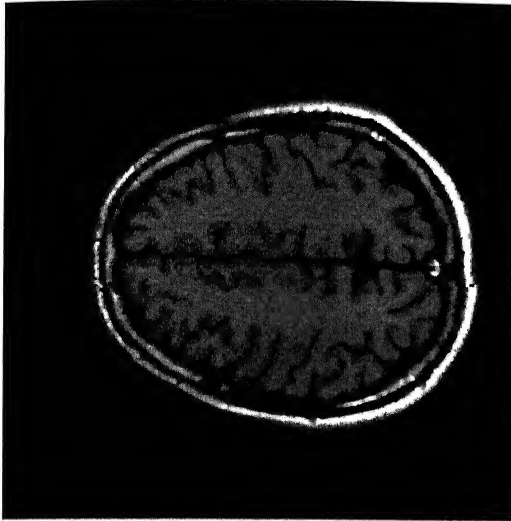
We consider test images at depth 8 bits per pixels. Image data is available in a 256 x 256 grid. Pixel values are available at spacing  $h = 1/256$ . Let the range of pixel values in the test images be normalized to take values from 0 to  $2^8 - 1$ . Let  $\{P_{ij}\}$  be the pixel data. The horizontal  $k$ -th differences  $\Delta_h^k$  for suitable  $k$  are calculated for the pixel data using the equation (3.1.2). After computing the matrix of differences (cf: 3.1.3) corresponding to the pixel data they are stored as follows:

- (i) The entries of difference matrix are signed integers. Let the entries of the difference matrix be denoted by  $\{d_{ij}\}$ . Let  $r$  be the smallest negative number in the difference matrix.
- (ii)  $d_{ij}^1 = d_{ij} + |r|$ , are computed for the difference matrix. This is done to avoid the storage of a sign bit, required for each entry in the difference matrix.
- (iii) The storage matrix along with the header containing the value of  $r$  is entropy encoded using Huffman variable length coding (cf: 1.2.2)

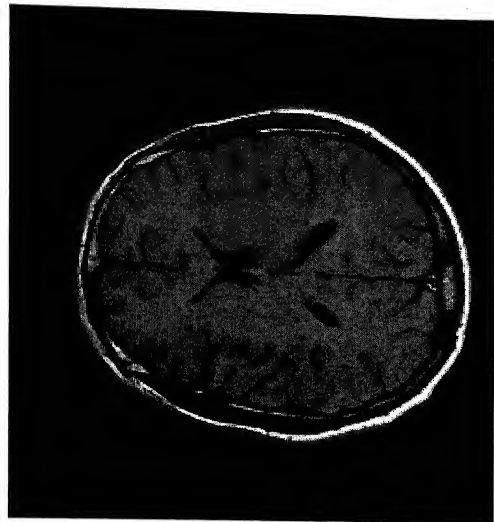
The resulting images after decompression are shown in Image 3.5.1. We note that the entropy of the storage matrix  $d_{ij}^1$  is less than the entropy of the actual image. Along with each of the decompressed image, the corresponding percentage compression, bit rate required for storage and the entropy of differences is given. We note that the performance of lossless compression algorithm 3.5.1 is comparable with JPEG coder at minimum loss levels (Image 3.5.2). The results of algorithm 3.5.1 applied on sharpened images are shown in Image 3.5.3 and Table 3.5.1. The smoothness estimates for the sharpened images are also shown in Table 3.5.1.

# Non-lossy Compression through Algorithm 3.2.1

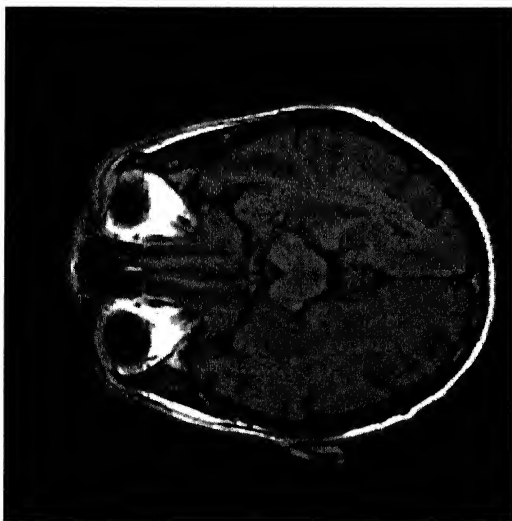
## Images after decompression



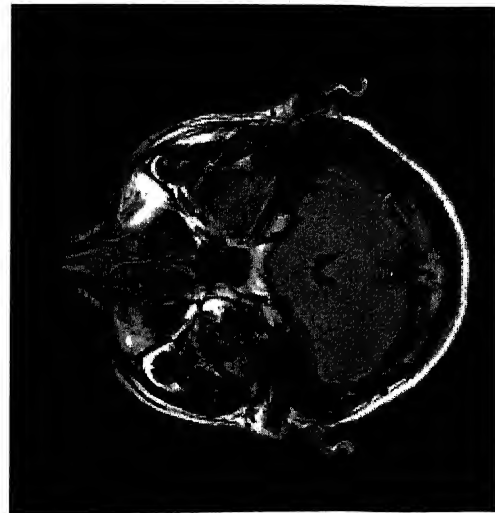
60.img decompressed  
P.C.: 40.30 % B.R : 4.77 bpp  
Entropy of differences : 4.682



64.img decompressed  
P.C.: 41.08 B.R.: 4.71 bpp  
Entropy of differences : 4.597

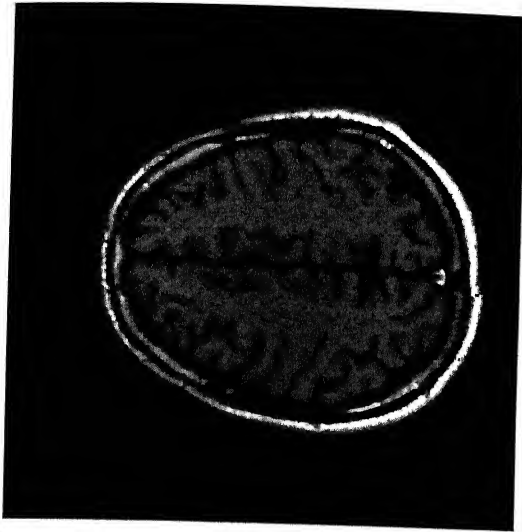


68.img decompressed  
P.C.: 38.83 % B.R.: 4.89 bpp  
Entropy of differences : 4.757

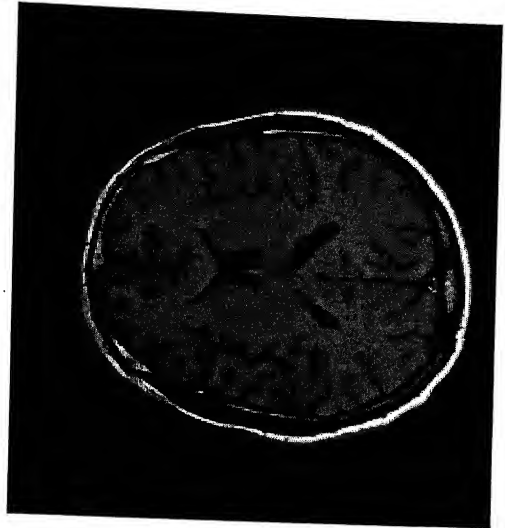


72.img uncompressed  
P.C.: 36.03 B.R.: 5.11 bpp  
Entropy of differences : 4.991

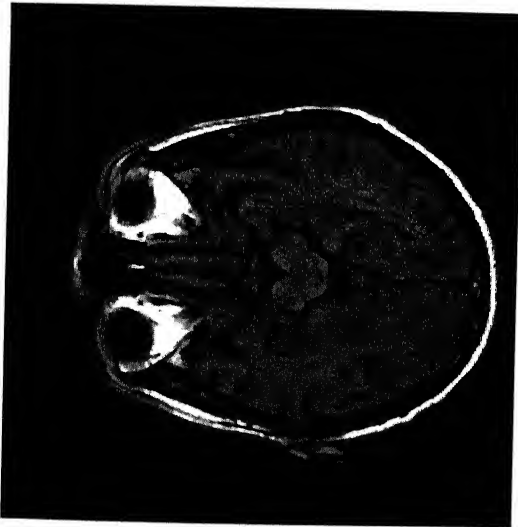
## Image compression through JPEG



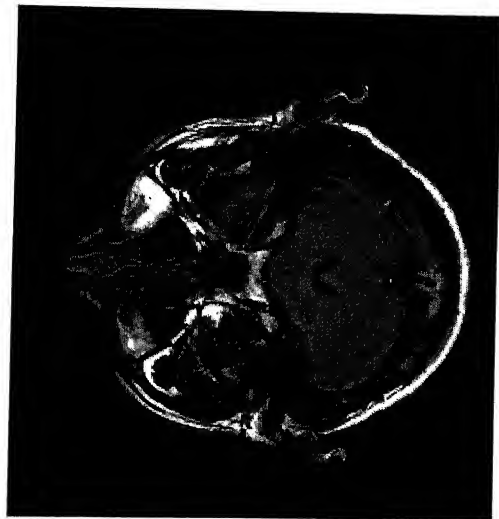
60.img.jpg  
Bit rate = 4.67 bpp



64.img.jpg  
Bit rate = 4.66 bpp



68.img.jpg  
Bit rate = 4.91 bpp



72.img.jpg  
Bit rate = 5.13 bpp

Image 3.5.2

## Compression of Sharpened images



img.36



img.3612



img.36123



img.361234

Image 3.5.3

The percentage compression and bit rates are presented in the following table.

### $\alpha$ -Estimates and Compression of sharpened images using Algorithm 3.2.1

Images	$\alpha$ - Estimates								P.C	Bit rate
	$l2-\omega_1$		$l2-\omega_2$		$l2-\omega_2$		$l2-\omega_2$			
	$\alpha$	C	$\alpha$	C	$\alpha$	C	$\alpha$	C		
Img.36	0.6608	12.71	0.9382	15.58	1.0911	24.081	1.1929	40.61	38.79	4.90
Img3612	0.5638	15.05	0.7879	20.28	0.9164	32.61	1.0049	56.16	35.89	5.12
Img.36123	0.4829	16.07	0.6568	22.97	0.7574	38.07	0.8281	66.79	34.05	5.28
Img.361234	0.4321	17.10	0.5841	25.25	0.6750	42.39	0.7410	74.88	33.89	5.29

Table 3.5.1

Table 3.5.1

Where C is the associated constant with  $\alpha$ . P.C is the percentage compression as defined in (chapter 2, sec. 2.6) and Bit rate is defined in (3.1.2).

From the table 3.2.1-3.2.2 we note that the smoothness of the test images decreases from img.60 to img.72. And we note that the compression is more for images, which are smoother. The main advantage of algorithm 3.5.1 is that it is non-lossy which is highly useful in medical imaging. In chapter 4 of the present thesis, we develop an algorithm based on reversible smoothing, which work well for the images under consideration

# CHAPTER IV: COMPRESSION THROUGH REVERSIBLE SMOOTHING

## 4.0 Introduction:

We have seen in the previous chapter that for smooth images the  $k$ -th differences are sufficiently small and can be used for compression. In the present chapter we propose and analyze a compression algorithm by reversible smoothing. We obtain a smooth version of the original image by convolution and store the smooth image by the  $k$ -th difference encoding as discussed in the previous chapter. At the time of reconstruction we first obtain the smooth image from its differences and an approximation to the original image is reconstructed from the smooth image by deconvolution. Let a given MRI image be,

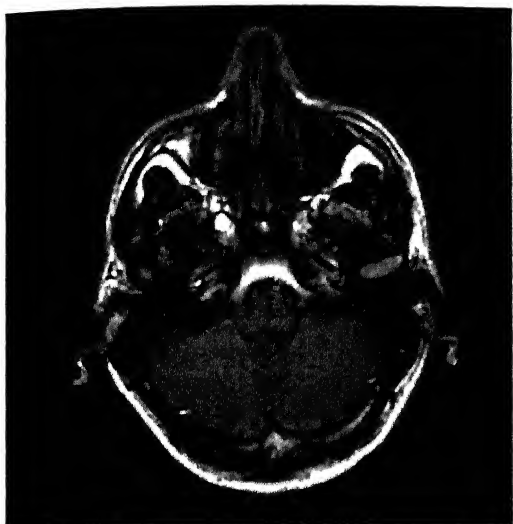
$$f(x, y) = \sum_{j,k} c_{j,k} e^{i(jx+ky)}. \quad (4.0.1)$$

We apply an invertible transformation on the Fourier coefficients to get a new function  $\tilde{f}$ . This transformation could be based on a known convolving function or otherwise. For the chosen convolving function,

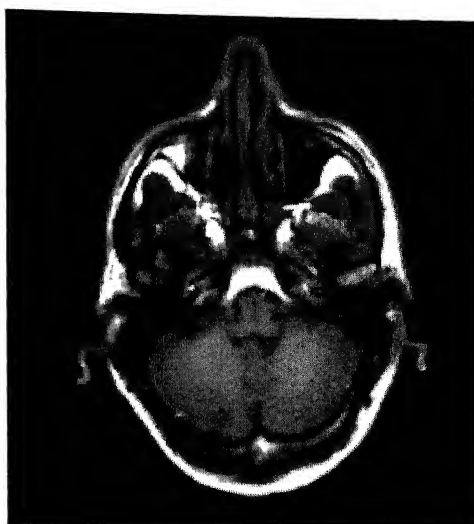
$$\tilde{f}(x, y) = \sum_{j,k} \tilde{c}_{j,k} e^{i(jx+ky)}. \quad (4.0.2)$$

$f$  is the original image and  $\tilde{f}$  is the smooth version of  $f$  obtained by convolving  $f$  with suitable approximation kernel. In practice  $\tilde{c}_{j,k}$ 's are obtained by multiplying  $c_{j,k}$ 's with  $\rho(k)$ 's, the filter values corresponding to the convolving function. Instead of storing  $f$  by algorithm (3.2.1) we store  $\tilde{f}$  by the same algorithm and at the time of decompression we get  $\tilde{f}$  first and from that  $f$  can be obtained by deconvolution. The process of compression through reversible smoothing is illustrated using the following example.

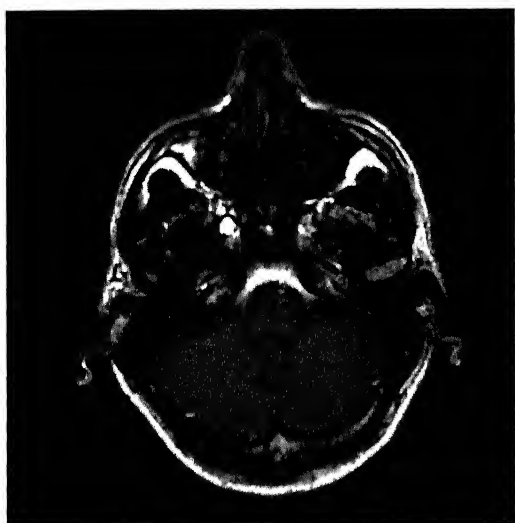




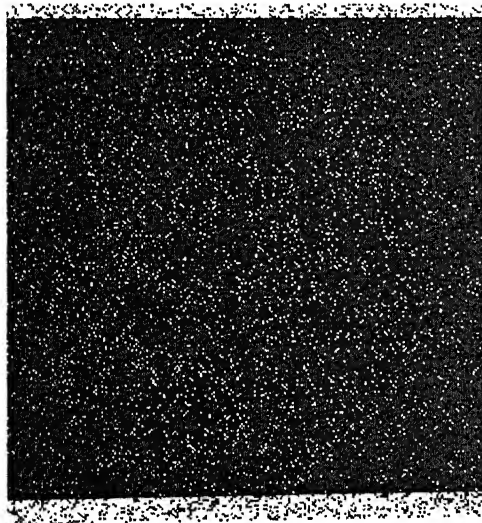
Original - Image 4.0.1  
P.C : 34.84% B.R.: 5.21 b



Smooth - Image 4.0.2  
P.C : 48.37 % B.R.: 4.13



Decompressed: Image 4.0.3  
Relative I1 error : 0.0618  
Relative I2 error : 0.0472  
Average I1 error : 0.0074  
Average I2 error : 0.0092



Error Image 4.0.4  
Absolute difference between Image  
4.0.1 and Image 4.0.3. Intensity  
increased 25 times

P.C – Percentage Compression, B.R-Bit Rate.

We consider four different approximation processes Fejer, Poisson, Jackson and Steklov means for smoothing the original images. The rest of the chapter is organized as follows, the four well known approximation processes listed above are reviewed and the corresponding filter coefficients are presented in the first section. In section two, we discuss the compression strategy and study the effect of smoothness in compression. Experimental implementations of the compression strategy are also carried out in section two. In section three a relation connecting sampling resolution and the corresponding depth is obtained. Section four is devoted to wavelet based compression methods.

### 4.1 Approximation Processes:

Let the original image  $f \in C_{2\pi}$ , and let it be represented by  $n$ -th partial sum of Fourier series,

$$S_n(f; x) = \frac{a_0}{2} + \sum_{k=1}^n a_k \cos kx + b_k \sin kx \quad (4.1.1)$$

where  $a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos kx dx$  and  $b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin kx dx$ .

It is well known that  $S_n(f)$  does not converge to  $f$  even for continuous functions. Several approximation processes are available in the literature (Zygmund, [79], Natanson [53]).

Most of them are available in the form of  $\frac{a_0}{2} + \sum_{k=1}^n \rho(k)(a_k \cos kx + b_k \sin kx)$ , where  $\rho(k)$ 's are the corresponding filter coefficients. A brief description of the filter coefficients corresponding to the approximation processes by using Fejer, Poisson, Steklov and Jackson kernels are as follows.

#### Fejer:

These are the arithmetic averages of the  $n$ -th partial sum of the Fourier series and are

obtained by convolving  $f$  with the Fejer's kernel  $\frac{\sin^2 \frac{nt}{2}}{\sin^2 \frac{t}{2}}$ .

$$F_n(f) = \frac{S_0^f + S_1^f + \dots + S_{n-1}^f}{n}$$

$$F_n(f; x) = \frac{1}{2\pi n} \int_{-\pi}^{\pi} \frac{\sin^2 \frac{n(t-x)}{2}}{\sin^2 \frac{(t-x)}{2}} f(t) dt$$

$$F_n(f; x) = \frac{a_0}{2} + \sum_{k=1}^n \left( \frac{n-k}{n} \right) (a_k \cos kx + b_k \sin kx) \quad (4.1.2)$$

The factor  $\left( \frac{n-k}{n} \right)$  acts as a weight on high frequency terms and kills the high frequency components thereby introducing some amount of smoothness in the image.

### Poisson:

Poisson sums are obtained by convolving  $f$  with the poisson kernel. For  $0 \leq r < 1$ , the Poisson kernel  $P(r, t)$  is given by

$$P(r, t) = 1 + 2 \sum_{j=1}^{\infty} r^j \cos jt = \frac{1-r^2}{1-2r \cos t + r^2}$$

$$P_n(f) = \frac{a_0}{2} + \sum_{k=1}^n r^k (a_k \cos kx + b_k \sin kx) \quad (4.1.3)$$

Here, the filter coefficients are generated by using  $r^k$  ( $0 < r < 1$ ). Smaller the value of  $r$  higher the smoothness, practically  $r$  cannot be too small because of the roundoffs that occur at the time of deconvolution, consequently we may loose some information, if the image is smoothened too much for the sake of compression. We show by experiments that there exists a practical choice of  $r$ , which yields good compression and acceptable roundoffs.

### Jackson:

Jackson's sums are obtained by convolving  $f$  with the Jackson kernel given by,

$$J_n(f; x) = \frac{1}{2\pi n} \left[ \frac{\sin \frac{nt}{2}}{\sin \frac{t}{2}} \right]^4 f(t) dt$$

$$J_n(f; x) = \frac{3}{2\pi n(2n^2 + 1)} \int_{-\pi}^{\pi} \left( \frac{\sin \frac{n(t-x)}{2}}{\sin \frac{t-x}{2}} \right)^4 f(t) dt \quad (4.1.4)$$

Rathore [61] has given an explicit construction of a sequence of linear trigonometric polynomial operators of the best approximation order as given by Jackson's theorems (see Natanson[53]). A particular case (Lemma 4.1.1) of his general theorem can be used to obtain the filter coefficients in the case of Jackson operators.

#### Lemma 4.1.1:

The following identity holds true

$$\left[ \frac{\sin \frac{nt}{2}}{\sin \frac{t}{2}} \right]^4 = L + \sum_{k=1}^{2n-2} l_k \cos kx,$$

$$\text{where } L = 4n^2 \left[ \frac{1}{2} + \sum_{k=1}^{n-1} \left( 1 - \frac{k}{n} \right)^2 \right] = \frac{4n^3 + 2n}{3} \text{ and}$$

$$l_k = \begin{cases} 2n(k+1) + \frac{k(k-1)(k+1)}{3} + \frac{2(n-k)(n-k-1)(2n+k-1)}{3} & 1 \leq k \leq n \\ \frac{(2n-k)(2n-k-1)(2n-k+1)}{3} & n < k \leq 2n-2 \end{cases}$$

The filter coefficient  $p(k)$ 's for the Jackson sums are obtained as follows,

$$\begin{aligned} J_n(f; x) &= \frac{3}{2\pi n(2n^2 + 1)} \int_{-\pi}^{\pi} \left( \frac{\sin \frac{n(t-x)}{2}}{\sin \frac{t-x}{2}} \right)^4 f(t) dt \\ &= \frac{3}{2\pi n(2n^2 + 1)} \left( L \int_{-\pi}^{\pi} f(t) dt + \sum_{k=1}^{2n-2} \frac{3l_k}{2n(2n^2 + 1)} \left[ \int_{-\pi}^{\pi} f(t) \cos kt \cos kx dt + \int_{-\pi}^{\pi} f(t) \sin kt \sin kx dx \right] \right) \\ &= \frac{3}{n(2n^2 + 1)} \frac{a_0}{2} + \sum_{k=1}^n \frac{3l_k}{2n(2n^2 + 1)} [a_k \cos kx + b_k \sin kx], \end{aligned} \quad (4.1.5)$$

where  $L$  and  $l_k$  are as in the Lemma 4.1.1.

**Steklov:**

The  $r$ -th Steklov (Timan [71]) means of  $f$  at a spacing  $h$ ,  $f_{h,r}$  are given by

$$f_{h,r}(x) = \frac{1}{2h} \int_{x-h}^{x+h} f_{h,r-1}(x) dx \quad \text{with } f_{h,0}(x) = f(x) \quad (4.1.6)$$

**Lemma 4.1.2:**

If (4.1.1) is the partial sum of the Fourier series of  $f$ , then for any integer  $r \geq 1$ ,

$$f_{h,r}(x) = \frac{a_0}{2} + \sum_{k=1}^n \left( \frac{\sin kh}{kh} \right)^r (a_k \cos kx + b_k \sin kx) \quad (4.1.7)$$

**Proof:**

$$\begin{aligned} \text{Let } r=1, \quad f_{h,1}(x) &= \frac{1}{2h} \int_{x-h}^{x+h} f(x) dx = \frac{1}{2h} \int_{x-h}^{x+h} \left( \frac{a_0}{2} + \sum_{k=1}^n (a_k \cos kx + b_k \sin kx) \right) dx \\ &= \frac{a_0}{2} + \sum_{k=1}^n \left( a_k \frac{1}{2h} \int_{x-h}^{x+h} \cos kx dx + b_k \frac{1}{2h} \int_{x-h}^{x+h} \sin kx dx \right) \\ &= \frac{a_0}{2} + \sum_{k=1}^n \left( \frac{\sin kh}{kh} \right) [a_k \cos kx + b_k \sin kx]. \end{aligned}$$

For  $r=1$ , (4.1.7) is true let us assume that it is true  $r=m-1$  and let us prove (4.1.7) for

$r=m$

$$\begin{aligned} f_{h,m}(x) &= \frac{1}{2h} \int_{x-h}^{x+h} f_{h,m-1}(x) dx = \frac{1}{2h} \int_{x-h}^{x+h} \left( \frac{\sin kh}{kh} \right)^{m-1} \left( \frac{a_0}{2} + \sum_{k=1}^n (a_k \cos kx + b_k \sin kx) \right) dx \\ &= \frac{a_0}{2} + \sum_{k=1}^n \left( \frac{\sin kh}{kh} \right)^{m-1} \left( a_k \frac{1}{2h} \int_{x-h}^{x+h} \cos kx dx + b_k \frac{1}{2h} \int_{x-h}^{x+h} \sin kx dx \right) \\ &= \frac{a_0}{2} + \sum_{k=1}^n \left( \frac{\sin kh}{kh} \right)^m (a_k \cos kx + b_k \sin kx). \end{aligned}$$

Hence 4.1.7 is true by induction on  $r$ .

## 4.2 Compression Algorithm:

We now present the compression algorithm based on reversible smoothing, let the given MRI image be of the form (4.0.1).

### 4.2.1 Compression:

**Step 1:** Fix the approximation process and calculate the filter coefficients  $\rho(k)$ 's.

**Step 2:** Compute  $\tilde{c}_{j,k}$ 's in (4.0.2) using the calculated  $\rho(k)$ 's and  $c_{j,k}$ 's and obtain smooth image  $\tilde{f}$ .

**Step 3:** Apply Algorithm (3.2.1) on  $\tilde{f}$ , and entropy encode the corresponding  $k$ -th differences using Huffman's variable length coding (1.2.1).

### 4.2.2 Decompression:

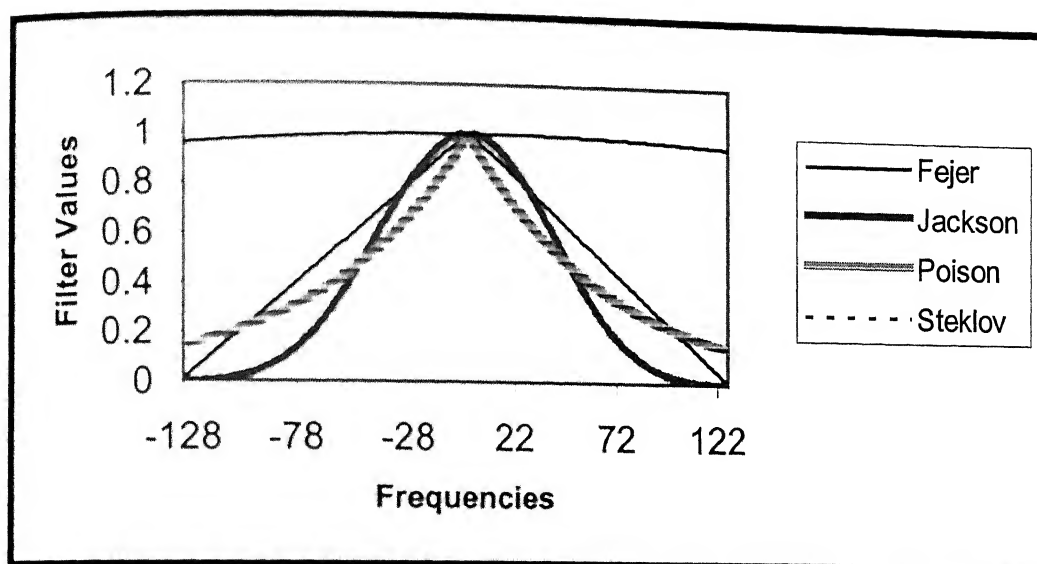
**Step 1:** Decode the variable length coded file by the procedure described in (1.2.1) and obtain the  $k$ -th differences of the smooth image.

**Step 2:** Obtain the smooth image from its stored  $k$ -th differences, using the relation (3.1.4).

**Step 3:** Reconstruct the image from the smooth image by deconvolution by the same convolving function used for smoothing.

The filter coefficients (windows) corresponding to the approximation processes listed above are as shown in the Graph 4.2.1. For the case of Poisson the value of  $r$  is taken to be 0.985.

## WINDOWS



Graph 4.2.1

### Experimental setup:

Here we smooth the 256x256 test images using the processes considered above. The test images have 256 gray scales, which take 8 bits per pixel for the original storage. The two measures we use for estimating the amount of compression are percentage compression and bit rate. Here the initial (original) storage for a 256 x 256 image at a bit rate 8 bpp is 65536 bytes. Although the transformation applied on the Fourier coefficient is invertible, by dividing the filtered coefficients with the respective filter values, it leads to some round off errors. So there is a practical range beyond which images cannot be smoothened for the sake of reversibility.

In case of Poisson approximation the value of  $r$  determines the imposed smoothness. Lower the value of  $r$  higher the smoothness obtained, but  $r$  cannot be too low. To estimate the useful range of  $r$  for the class of images under consideration, we measure the error after decompression. Four different quantities are used (cf: 2.6), relative l1 (rl1), relative l2 (rl2), average l1 (al1) and average l2 (al2). Besides these error measures to analyze the performance of the compression algorithm we introduce a measure of gain, it is defined as follows,

$$r11 \text{ gain} = \frac{\text{Compression Ratio}}{\text{relative l1 error after decompression}} \quad (4.2.1)$$

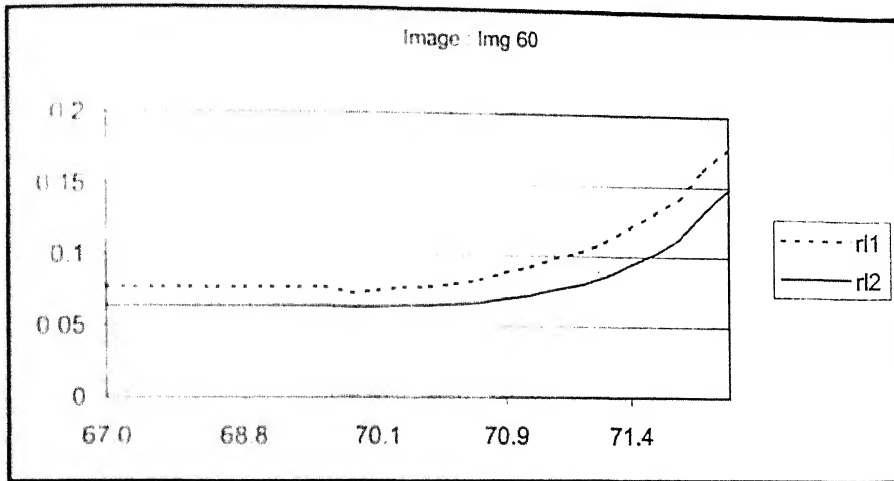
$$r12 \text{ gain} = \frac{\text{Compression Ratio}}{\text{relative l2 error after decompression}} \quad (4.2.2)$$

As  $r$  decreases from 1.0, the compression ratio increases as the smoothness of the image increases. The relative error decreases up to some point and then increases, the point where relative gain attains maximum is the region of interest.

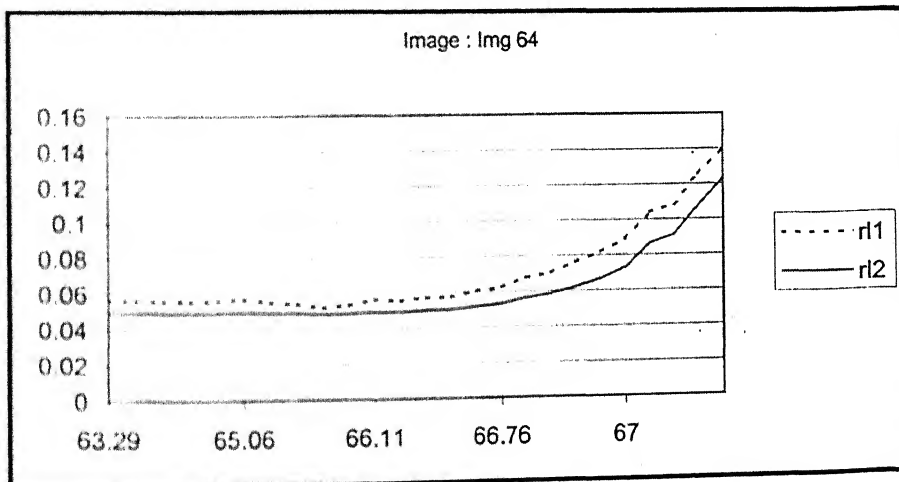
Result of the compression algorithm 4.2.1 using convolution through Poisson kernel is shown in Graph 4.2.2-4.2.5. Variations of compression with respect to smoothness in terms of relative gain are shown in Graph 4.2.6-4.2.9. It is seen that for the convolution of pixel data using Poisson kernel, the value of  $r = 0.985$  works well for the test images under consideration. Images after decompression for the case of Poisson, Fejer and Steklov are shown in Image 4.2.1-4.2.3 and 4.2.5. Sharpened images (Test image set 2) are compressed using the algorithm 4.2.1 and the results are presented in image 4.2.4. Results indicate that for visual usage of images compression strategy 4.2.1 based on Poisson process is better suited. The compression ratio attained for the test images are higher when compared to other methods studied in the previous chapters of this thesis.



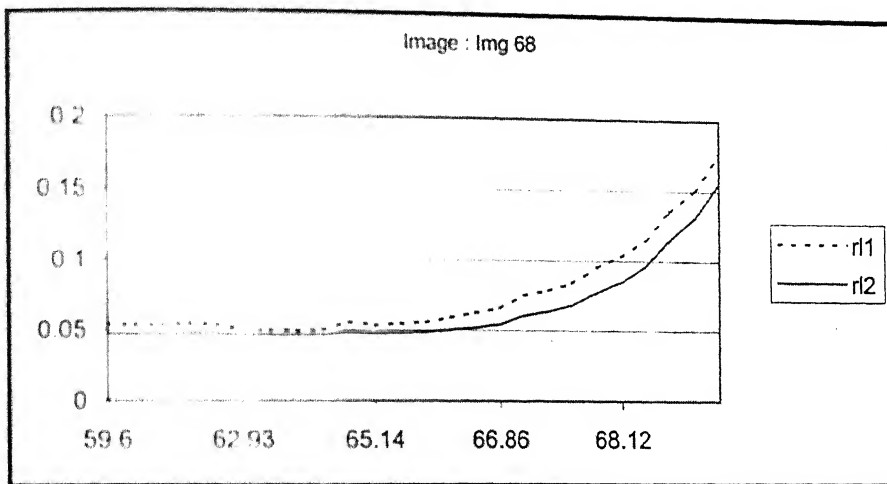
## Percentage Compression Vs Relative Error (Poisson Process)



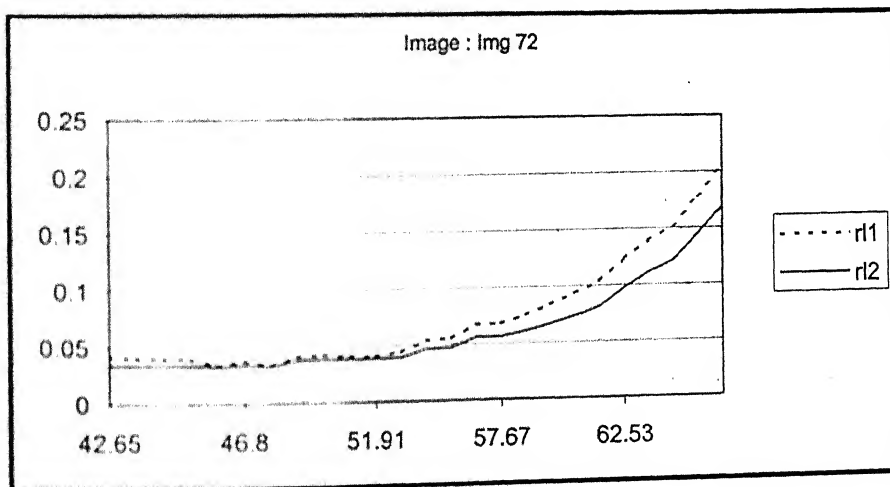
Graph 4.2.2



Graph 4.2.3

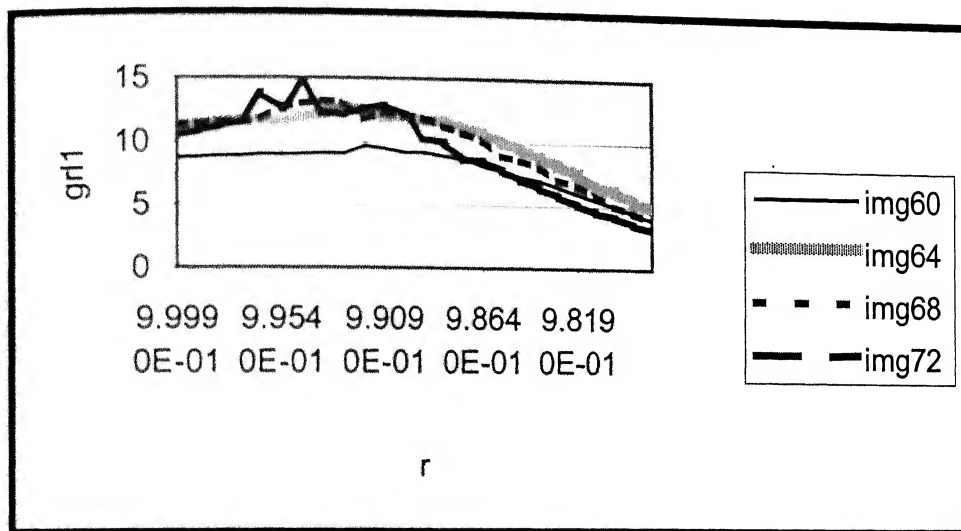


Graph 4.2.4

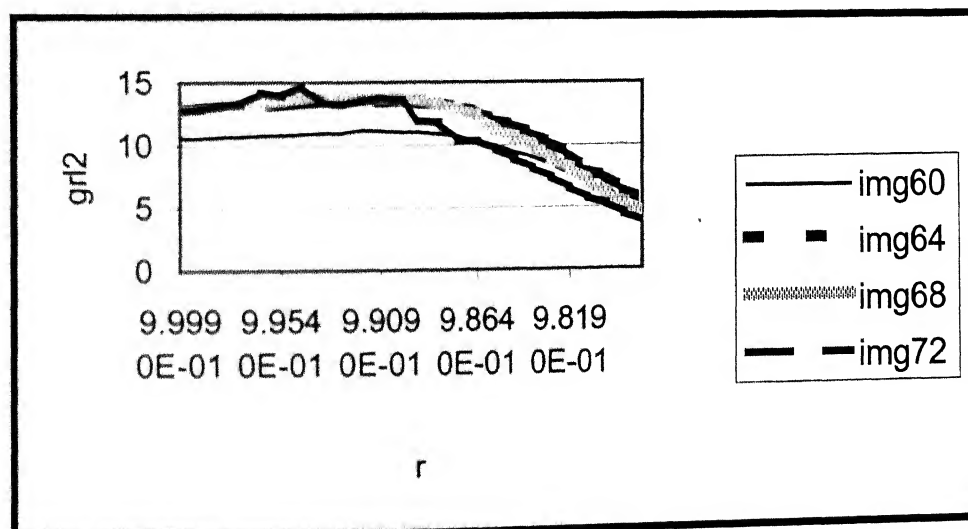


Graph 4.2.5

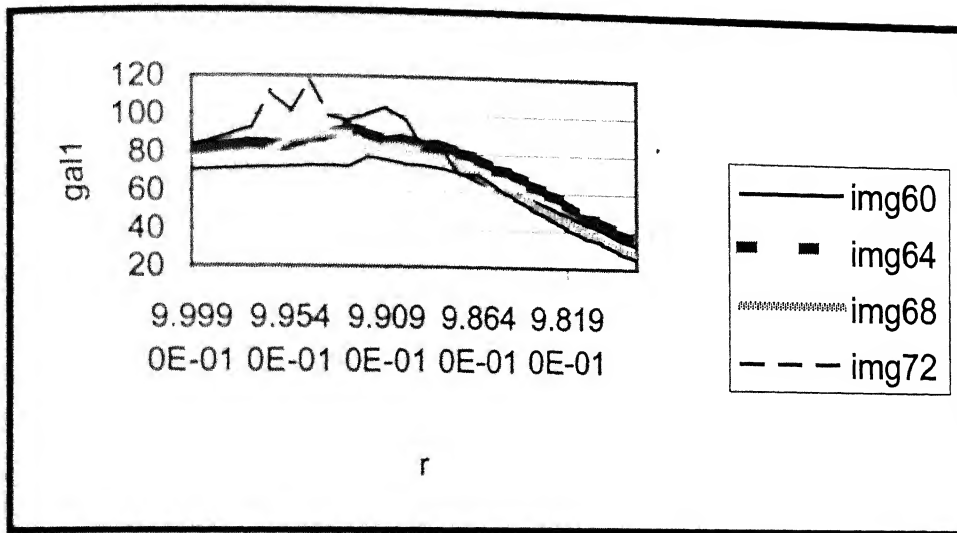
## Variation of gain with respect to smoothness (Poisson process)



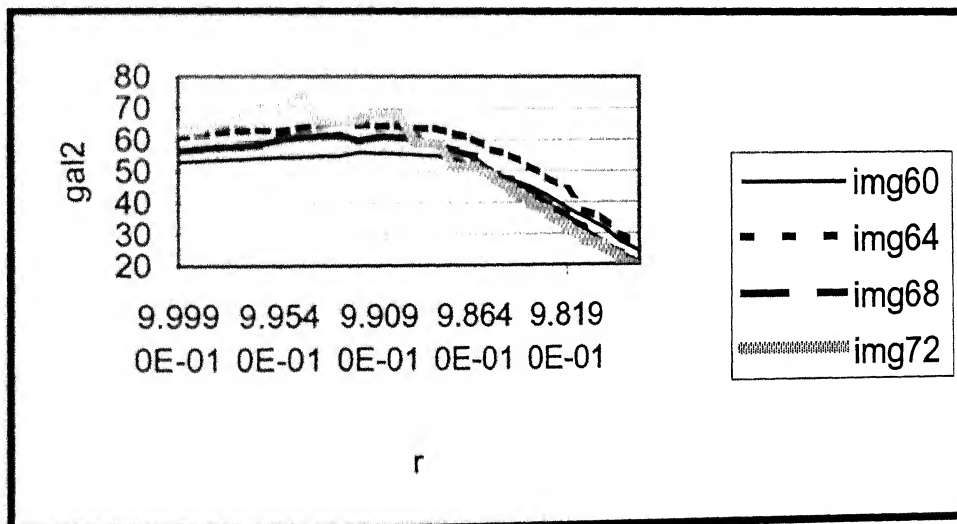
Graph 4.2.6



Graph 4.2.7



Graph 4.2.8



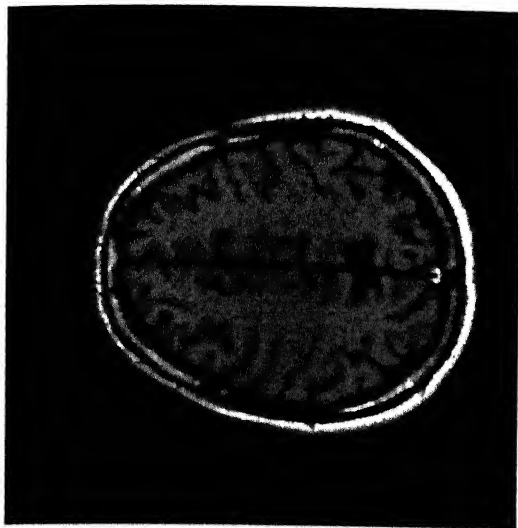
Graph 4.2.9

In case of Poisson approximation, the graph of Smoothness( $r$ ) Vs gain indicates that there exists a point at which the gain is maximal. The round off error at that value of  $r$  is admissible (there is no visual loss of information in the images obtained after decompression). As  $r$  decreases, the gain increases initially but after some stage the round off errors in deconvolution dominates and hence the gain starts decreasing.

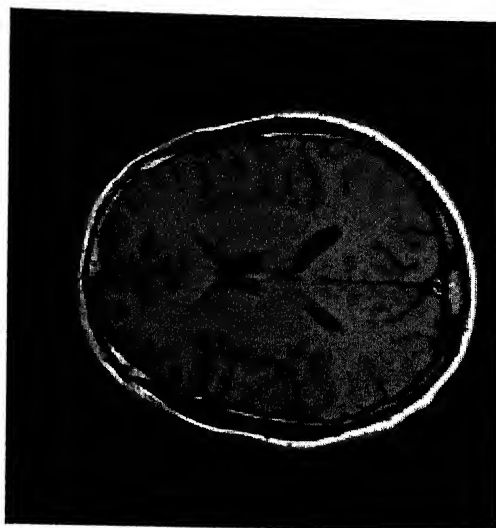
Decompressed images corresponding to maximal gains for the test images are shown in Image 4.2.1. In the case of Fejer approximation, the  $k$ -th Fourier coefficient  $\hat{f}(k)$  is multiplied by a quantity  $\frac{N-|k|}{N}$  (filter coefficients) where  $N$  is the highest frequency, (in our case it is 128) and  $k$  runs from  $-127$  to  $128$ . During the process of decompression, the Fourier coefficients calculated from the smooth image are divided by the filter coefficients. When the value of  $k$  is 128, we see that the multiplier becomes zero and causes problems at the time of decompression. To avoid this, we take  $N=129$  and compute the filter coefficients. We observe that for  $N = 129$ , the amount of smoothness imposed by filtering is more. Although, the smooth image obtained by using,  $N=129$  gives better compression, it lacks reversibility. The decompressed image 4.2.2 (corresponding to  $N=129$ ) in this process shows up deterioration. As the value of  $N$  increases, the visual deterioration starts decreasing. For all practical purposes, we use the value of  $N = 130$ , which strikes a good balance between compression and reversibility. The graph of  $N$  vs decompression error is shown in Graph 4.2.10-4.2.11 and the images after decompression for various values of  $N$  are presented in Image 4.2.2-4.2.3. In case of the Steklov approximation, not much smoothing occurs. The Steklov window as shown in Graph-4.2.3 is almost horizontal and all the filter values are near 1. The decompressed images are comparable to that of images obtained in Poisson's process.

In comparison with Fejers and Steklov, the Poisson approximation process works better for the test images considered here. The additional advantage of using the Poisson kernel is that the value of  $r$  can be suitably adjusted, so for different values of  $r$  we get different compression. We present the images after decompression through these processes along with their compression percentage and the respective errors.

## Poisson process - Images after decompression



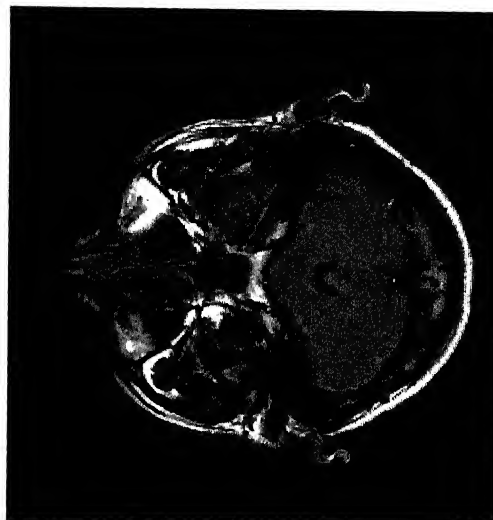
Percentage compression : 69.94  
Bit rate : 2.40 bpp Size : 19697 bytes



Percentage compression : 66.66  
Bit rate : 2.67 bpp Size : 21852 bytes



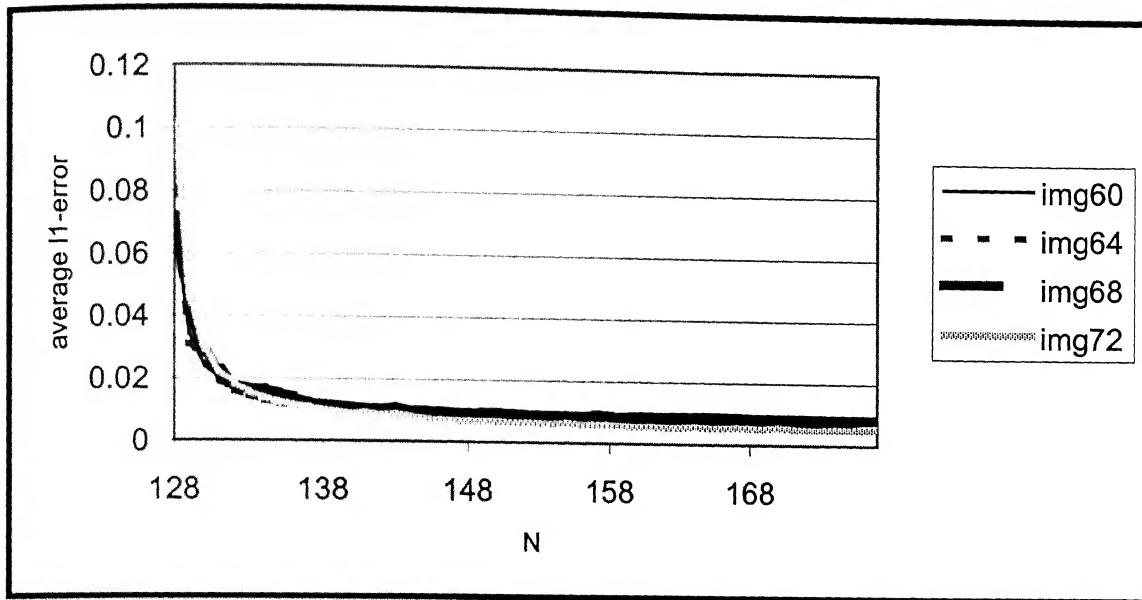
Percentage compression : 66.46  
Bit rate : 2.68 bpp Size : 21982 bytes



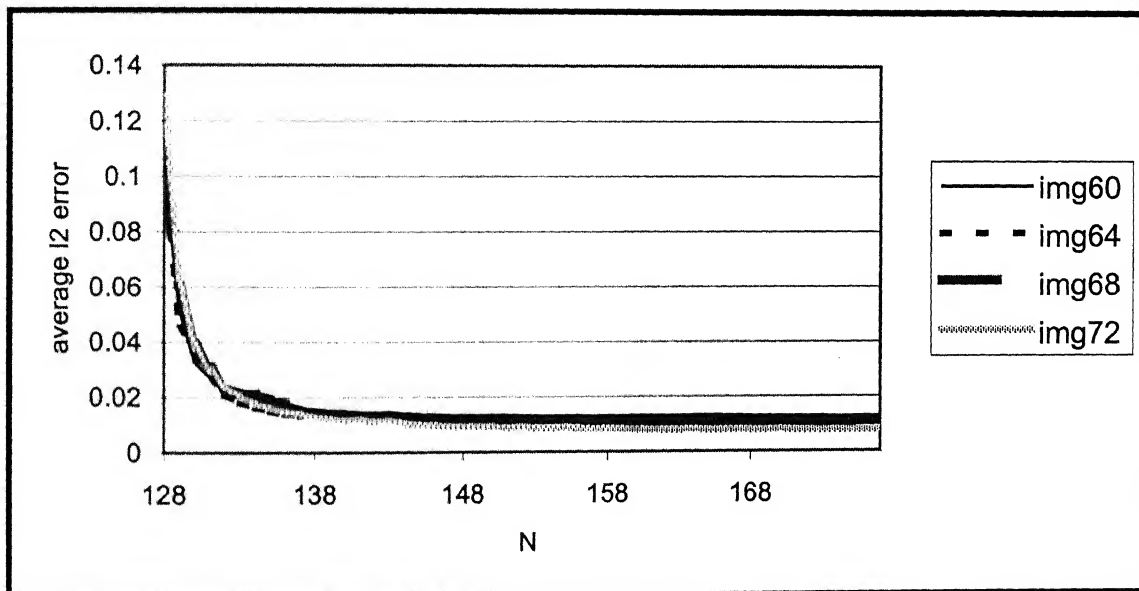
Percentage compression : 59.33  
Bit rate : 3.25 bpp Size : 26656 bytes

Image 4.2.1

## Variation of Smoothness with N in Fejer's smoothing

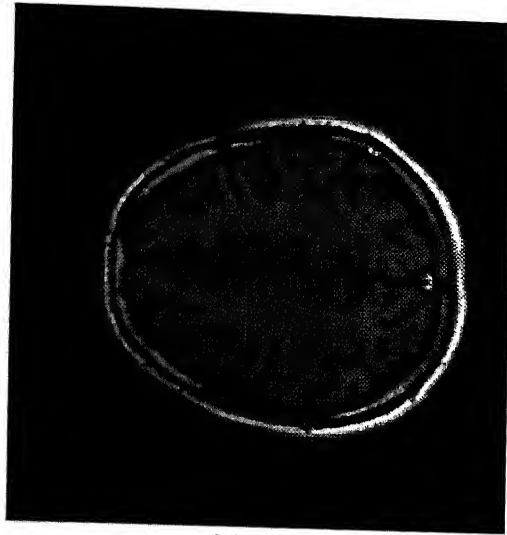


Graph 4.2.10

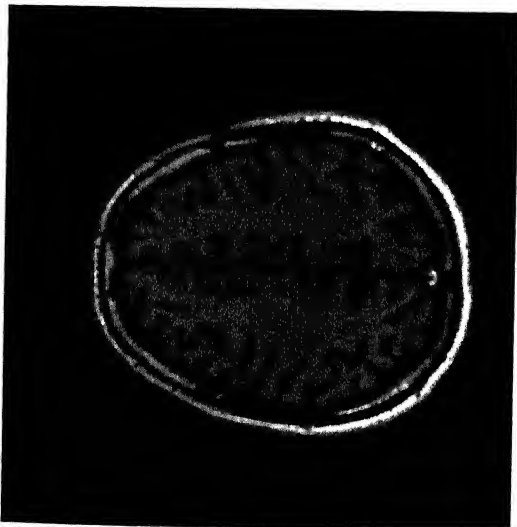


Graph 4.2.11

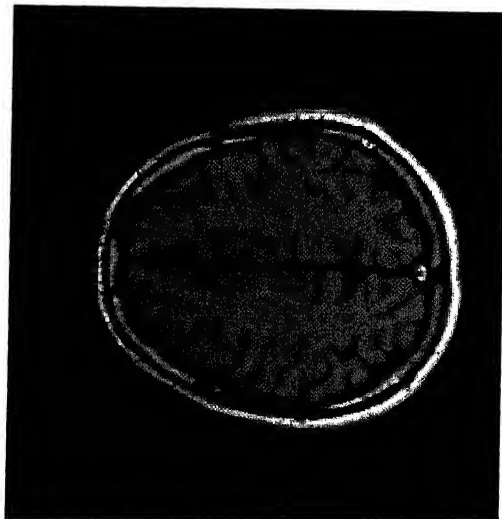
# Fejer smoothing - Images after decompression for various values of $N$



$N=129$



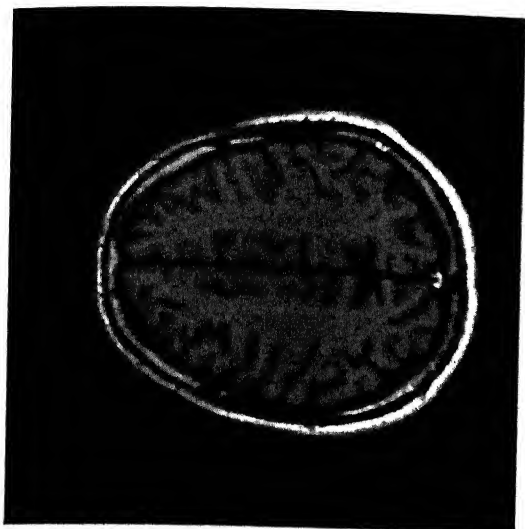
$N=130$



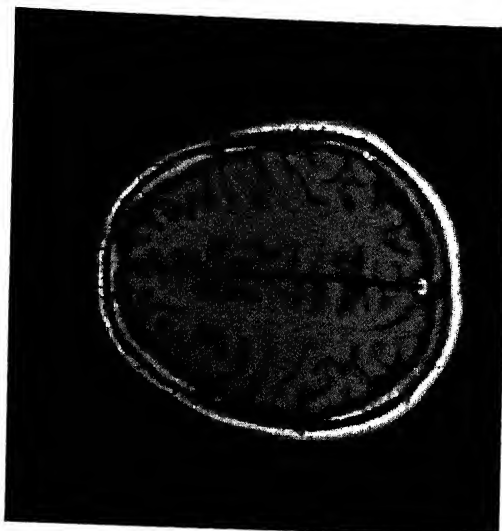
$N=131$

Image 4.2.2

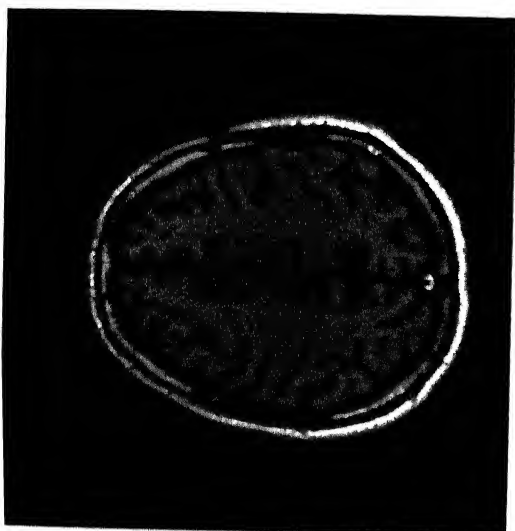




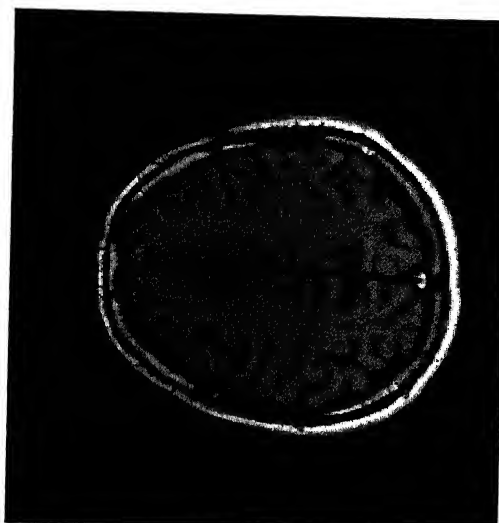
N=132



N=133



N=134



N=135

Image 4.2.3

# COMPRESSION OF SHARPENED IMAGES (Using test image set : 2) POISSON PROCESS-ALGORITHM 4.2.1

Images after decompression



$r = 0.9846$  P.C : 63.94 % B.R.: 2.88  
Relative error rl1: 0.0567



$r = 0.9837$  P.C : 61.61 % B.R.: 3.07  
Relative error rl1: 0.0628



$r = 0.9828$  P.C : 60.62 % B.R.: 3.15  
Relative error rl1: 0.0749



$r = 0.9819$  P.C : 60.53 % B.R.: 3.16  
Relative error rl1: 0.0811

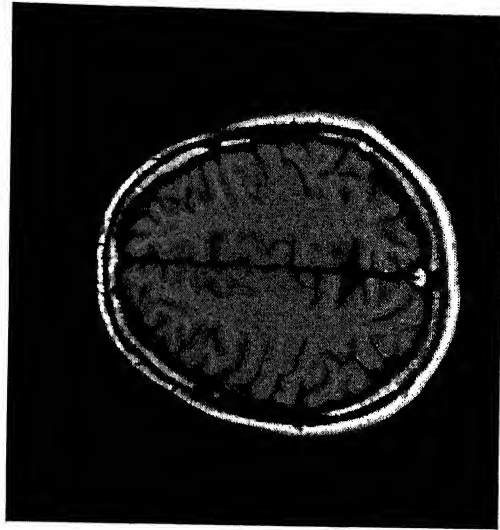
In the above images,  $r$  is the corresponding Poisson kernel parameter.

Image 4.2.4

## Steklov Smoothing



Steklov smooth Image corresponds to  
img 60 Compression : 67.04% and Bit  
rate = 2.64 bpp



Decompressed from Steklov smooth image  
Relative l1 error = 0.076: Relative l2  
error = 0.063: Average l1 error = 0.00095:  
Average l2 error = 0.013

Image 4.2.5

In the next section we study the effect of smoothness in sampling. The resolution and depth of a digital image are related to the smoothness of the image.

### 4.3 Non Redundant Sampling:

The depth and the resolution of a digital image determine the initial storage requirements. If the image is sufficiently smooth, then for a required depth, the resolution can be coarse. The extreme example is a constant image which requires just one pixel information to reconstruct the entire image. Detailed images require fine resolution. The quantity modulus of smoothness plays an important role in fixing the resolution for a given depth. For a given depth, the resolution and smoothness measured in terms of modulus of smoothness can be related.

**Definition 4.3.1:** The Space  $X^{k+1,\alpha}$  is the set of functions whose  $\omega_{k+1}(f,h)$  is of the order of  $h^\alpha$ , where  $0 < \alpha < k+1$ .

$$X^{k+1,\alpha} = \{ f : \omega_{k+1}(f,h) = O(h^\alpha) \}.$$

Let  $L_{k,h}^f$  be the  $k$ -th degree interpolation polynomial (cf: 3.3.8) of  $f$ , the analog intensity distribution function.  $L_{k,h}^f$  is constructed using the  $k$ -th differences stored as per the compression algorithm (3.2.9).  $L_{k,h}^f$  is said to be a  $b$ -perfect decomposition of  $f$ , if the first  $b$ -bits of  $L_{k,h}^f$  match with that of  $f$ . Resolution  $m$  ( $h = 2^{-m}$ ), and depth  $b$  bits per pixel are two main parameters of a digital image. In this section, we address the question of what is the minimum resolution required to reconstruct the image at any arbitrary point in the domain under consideration, such that the reconstructed image is  $b$ -perfect. Analogously, the question of what is the maximum possible depth that can be attained at any arbitrary point in the domain of the image, by using a digital image at a resolution  $m$ . The pixel value at any arbitrary point is evaluated using  $L_{k,h}^f$  (with  $h = 2^{-m}$ ). The following theorem gives the relation between depth and resolution of a digital image in terms of the modeled smoothness.

**Definition 4.3.2 (Maximum attainable depth  $b_m$  using  $L_{k,2^{-m}}^f$ ):**

Let  $f$  be the original image and  $L_{k,2^{-m}}^f$  be a reconstruction of  $f$  from the pixel data at a resolution  $m$  (spacing  $= 2^{-m}$ ), by using a local Lagrange's interpolation polynomial of degree  $k$ . Then, the maximum attainable depth is the quantity  $b_m$  such that,

$$2^{-b_m-2} \leq \|f - L_{k,2^{-m}}^f\|_\infty \leq 2^{-b_m-1}.$$

**Theorem 4.3.1:**

Let  $f$  be the original image and  $L_{k,2^{-m}}^f$  be the reconstruction. Let  $b_m$  be the depth associated with  $m$ . Then the following holds:

$$f \in X^{k+1,\alpha} \text{ if and only if there exists a constant } C \text{ such that } \frac{b_m}{m} \geq \alpha - \frac{C}{m}.$$

**Proof:**

Let  $f \in X^{k+1}$ . By theorem 3.4.2 we have,

$$\begin{aligned} \|f - L_{k,2^{-m}}^f\| &\leq C_1 \omega_{k+1}(f; 2^{-m}) \\ &\leq 2^{C_2 - m\alpha} \end{aligned} \quad [\because f \in X^{k+1,\alpha}]$$

$$2^{-b_m-2} \leq \|f - L_{k,2^{-m}}^f\| \leq 2^{C_2 - m\alpha}$$

$$\Rightarrow m\alpha - b_m \leq C \quad [\text{By definition 4.3.2}]$$

$$\therefore \frac{b_m}{m} \geq \alpha - \frac{C}{m}$$

Conversely, let  $C$  be a constant such that  $\frac{b_m}{m} \geq \alpha - \frac{C}{m}$  holds.

$$\Rightarrow m\alpha - b_m \leq C$$

$$\frac{\omega_{k+1}(f, 2^{-m})}{2^{-m\alpha}} \leq \frac{C_1 \|f - L_{k,2^{-m}}^f\|}{2^{-m\alpha}} \quad [\text{By theorem 3.4.3}]$$

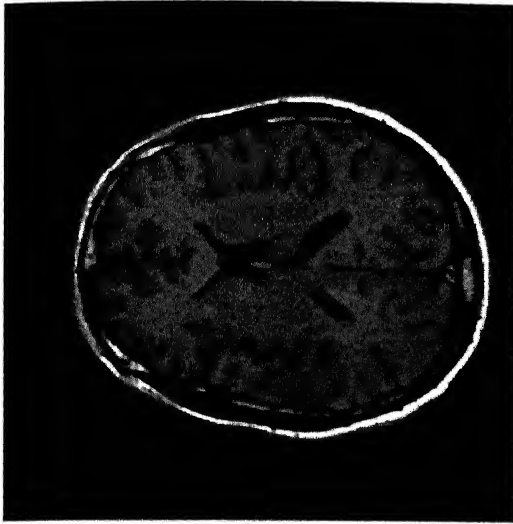
$$\leq \frac{C_1 2^{-b_m-1}}{2^{-m\alpha}} \quad [\text{By definition 4.3.2}]$$

$$\leq C_1 2^{C-1} \leq C_2 \text{ (say)}$$

$$\Rightarrow \left\{ \frac{\omega_{k+1}(f, 2^{-m})}{2^{-m\alpha}} \right\}_m \text{ is uniformly bounded sequence. } \therefore f \in X^{k+1,\alpha}.$$

From the above theorem we infer that, if for an image, the value of  $\alpha$  is high then smaller value of  $m$  is sufficient for b-perfect reconstruction. This shows that for smooth images the storage requirement is less. This is in accordance with the experimental result (4.3.1). As the depth of test image decreases the image architecture starts deteriorating. We present a test image at various depths and we see that the deterioration is sharp as the number of bits per pixel (bpp) is reduced.

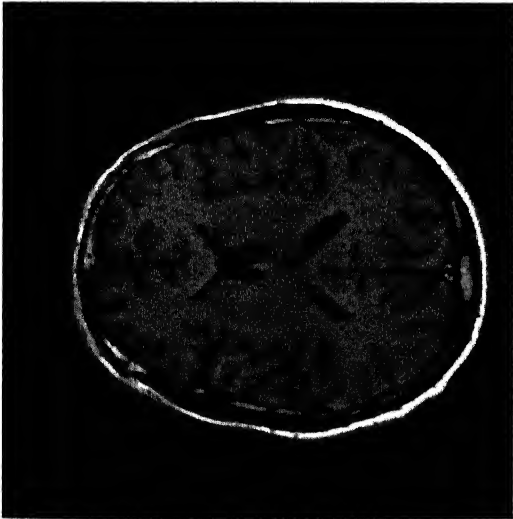
### Images at various depth



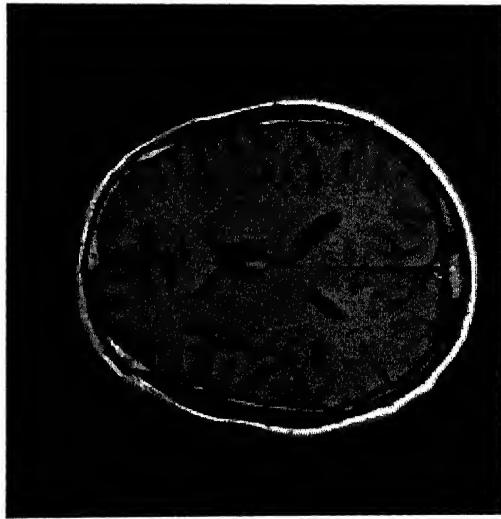
Depth 7 bpp



Depth 6 bpp



Depth 5 bpp



Depth 4 bpp

Image 4.3.1

In the next section we consider another important class of compression method, based on wavelets.

#### 4.4 Wavelet Basics:

In this section, we present some wavelet preliminaries, wavelet based compression algorithm and experimental results. These are used for comparison with the results obtained through methods described in the previous section.

Wavelets are elements generated from one single function  $\psi$  by its dilation and translations.

$$\psi^{a,b}(t) = |a|^{-1/2} \psi\left(\frac{t-b}{a}\right),$$

where  $\psi$  is the mother wavelet satisfying  $\int_{-\infty}^{\infty} \psi(x) dx = 0$ . In addition,  $\psi$  has to satisfy

admissibility condition, which is  $\int_{-\infty}^{\infty} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty$ . High frequency wavelets correspond

to a  $< 1$  or narrow width, while low frequency wavelets have a  $> 1$  or wider width.

The main objective of the wavelet transform is to represent any arbitrary function as a superposition of wavelets. Such a superposition decomposes  $f$  into different scale levels, where each level is then further decomposed with a resolution adapted to the level. Let  $f \in L_2(\mathfrak{R})$ . A decomposition of  $f$  can be achieved by writing  $f$  as an integral over  $a$  and  $b$  of  $\psi^{a,b}$ ,

$$Wf(a,b) = |a|^{-1/2} \int_{-\infty}^{\infty} f(x) \psi\left(\frac{x-b}{a}\right) dx. \quad (4.4.1)$$

This refers to the continuous parameter wavelet transform, the dilation and translation parameters  $a$  and  $b$  run over  $\mathfrak{R}^+$ . In most practical applications, discrete parameter wavelet transform is used. It is obtained by restricting the parameters  $a$  and  $b$  only to discrete values,  $a = a_0^m$ ,  $b = nb_0 a_0^m$ ,  $m, n \in \mathbb{Z}^+$ ,  $a_0 > 1$ ,  $b_0 > 0$ , fixed.

$$W_{m,n}(f) = C_{m,n}(f) = a_0^{-m/2} \int_{-\infty}^{\infty} f(t) \psi(a_0^{-m}t - nb_0) dt. \quad (4.4.2)$$

Then the wavelet decomposition is given by,

$$f = \sum_{m,n} C_{m,n}(f) \psi_{m,n} \quad (4.4.3)$$

$$\psi_{m,n}(t) = a_0^{-m/2} \psi(a_0^{-m}t - nb_0).$$

For  $a_0 = 2$  and  $b_0 = 1$ , there exist special choices of  $\psi$  such that  $\psi_{m,n}$  constitute an orthonormal basis and

$$C_{m,n}(f) = \langle \psi_{m,n}, f \rangle = \int_{-\infty}^{\infty} f(x) \psi_{m,n}(x) dx. \quad (4.4.4)$$

Different bases of this nature were constructed by Stromberg[70], Meyer[52], Battle[4] and Daubechies [15]. Mallat[49] invented a technique called Multi resolution analysis to the wavelet bases, in image analysis, and has given a fast computation algorithm.

In Multi resolution analysis there are two functions, the mother wavelet  $\psi$  and the scaling function  $\phi$ . The dilated and translated versions of the scaling function  $\phi_{m,n}(x) = 2^{-m/2} \phi(2^{-m}x - n)$ . For a fixed  $m$ , the  $\phi_{m,n}$  are orthonormal. The space spanned by  $\phi_{m,n}$  is denoted by  $V_m$  and the successive approximation spaces are of the form  $\dots V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \dots$ , each with resolution  $2^m$ . For each  $m$ , the  $\psi_{m,n}$  span a space  $W_m$  which is the orthogonal complement in  $V_{m-1}$  of  $V_m$ . The coefficients therefore describe the information lost when going from an approximation of  $f$  with resolution  $2^{m-1}$  to the coarser approximation with the resolution  $2^m$ . These ideas are translated into the following algorithm for the computation of  $C_{m,n}(f)$  (Daubechies [14], has more details):

$$\begin{aligned} C_{m,n}(f) &= \sum_k g_{2n-k} a_{m-1,k}(f) \\ a_{m,n}(f) &= \sum_k h_{2n-k} a_{m-1,k}(f). \end{aligned} \quad (4.4.5)$$

Where  $g_l = (-1)^l h_{-l+1}$  and  $h_n = 2^{1/2} \int \phi(x-n) \phi(2x) dx$ . If the function  $f$  is given in sampled form, then one can take these samples for the highest order resolution approximation coefficients  $a_{0,n}$  and the expressions for  $C_{m,n}$  and  $a_{m,n}$  describes a sub-band coding algorithm on these sampled values, with low pass filter  $h$  and high pass filter  $g$ . The reconstruction can be done using

$$a_{m-1,l}(f) = \sum_n \left[ h_{2n-l} a_{m,n}(f) + g_{2n-l} C_{m,n}(f) \right]. \quad (4.4.6)$$



#### 4.4.1 Compression Algorithm and storage bounds:

Here we present the wavelet based compression strategy (R.A. DeVore et al., [17]). The two major steps involved in compression using wavelet transform coding is to obtain the transform coefficients and then quantization of the transform coefficients. The quantized transform coefficients are stored instead of the pixel data.

**Step 1:** For the chosen wavelet basis, the coefficients  $C_{m,n}f$  as in 4.4.3 are calculated using the pixel data.

**Step 2:** Choose a positive integer  $N$ , and a number  $0 < \alpha \leq \frac{1}{2}$ . Let  $\tau = \left(\alpha + \frac{1}{2}\right)^{-1}$ . The quantized coefficients  $\tilde{C}_{m,n}(f)$ , are chosen such that they satisfy the following:

$$(i) \quad \left| C_{j,k}(f) - \tilde{C}_{j,k}(f) \right|^\tau \leq \frac{1}{N},$$

$$(ii) \quad |C_{j,k}| \leq |\tilde{C}_{j,k}|, \text{ and}$$

$$(iii) \quad |\tilde{C}_{j,k}| = 0 \text{ if } |C_{j,k}| \leq \frac{1}{N}.$$

**Step 3:** The quantized coefficients  $\tilde{C}_{m,n}(f)$ 's are stored using the entropy encoder based on Huffman coding (cf: 1.2.1).

Using the bounds on the  $l_p$  norm of the wavelet coefficients, in terms of the semi norm on Besov spaces, a bound on storage requirements can be obtained. The  $l_p$  bound on wavelet coefficients for wavelets arising from a multi resolution analysis with the scaling function and the wavelet satisfying the following conditions are available in Wojtaszczyk [75]. Here we state it as a lemma. The conditions on scaling function and the wavelets are

$$(i) \quad |\phi(x)| \leq C(1 + |x|)^{-A}$$

$$(ii) \quad |\phi'(x)| \leq C(1 + |x|)^{-A}$$

$$(iii) \quad |\psi(x)| \leq C(1 + |x|)^{-A} \quad \text{for some } A > 3$$

**Lemma 4.4.1:** Let  $\alpha$ ,  $0 < \alpha \leq \frac{1}{2}$  be given, and put  $\tau = \left(\alpha + \frac{1}{2}\right)^{-1}$ . Then there exists a constant  $C$  such that.

$$\sum_{j,k} \left| \langle f, \psi_{j,k} \rangle \right|^\tau \leq C |f|_{\tau,\alpha,\tau}^\tau,$$

$$\text{where, } |f|_{\tau,\alpha,\tau} = \begin{cases} \left( \int_0^\sigma \left[ t^{-\alpha} \omega(f;t)_\tau \right]^\tau \frac{dt}{t} \right)^{\frac{1}{\tau}} & 1 \leq \tau < \infty \\ \sup_{0 < t < \sigma} t^{-\alpha} \omega(f;t)_\tau & \tau = \infty \end{cases}$$

**Theorem 4.4.1 (Storage Bound):**

Let  $\eta$  be the number of non-zero coefficients  $\tilde{C}_{i,j}$  obtained after the application of Algorithm 4.4.1 for compression. Then  $\eta$  satisfies the following bound,

$$\eta \leq CN |f|_{\tau,\alpha,\tau}^\tau.$$

**Proof:**

$C_{j,k} = \langle f, \psi_{j,k} \rangle$  and the compressed coefficients  $\tilde{C}_{j,k}$  satisfy  $|\tilde{C}_{j,k}| \leq |C_{j,k}|$  as per the algorithm. We have,

$$\begin{aligned} \frac{\eta}{N} &\leq \sum_{j,k} |\tilde{C}_{j,k}| \leq \sum_{j,k} |C_{j,k}| = \sum_{j,k} \left| \langle f, \psi_{j,k} \rangle \right| \leq \sum_{j,k} \left| \langle f, \psi_{j,k} \rangle \right|^\tau \\ &\leq C |f|_{\tau,\alpha,\tau}^\tau \quad \quad \quad [\text{By lemma 4.4.1}] \end{aligned}$$

Therefore we have,

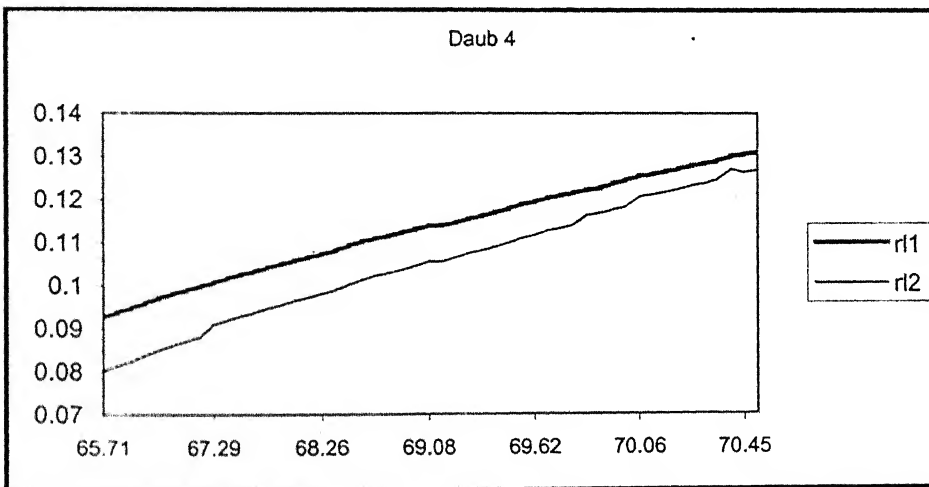
$$\eta \leq CN |f|_{\tau,\alpha,\tau}^\tau.$$

#### 4.4.2 Experiments:

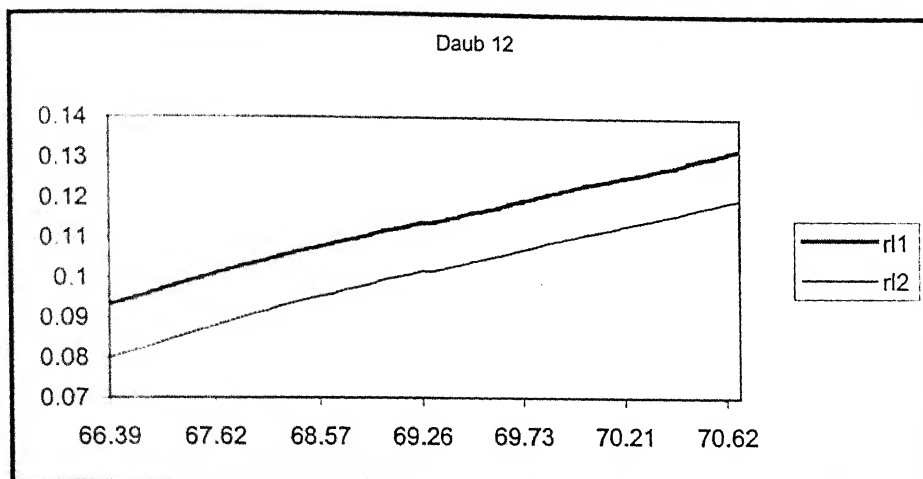
This is experimentally implemented on the test images. For experiments we use three wavelets, Daub4, Daub12 and Daub20, developed by Daubechies[14]. The numbers 4,12

and 20 represents the number of filter coefficients. We begin with an image which is coded at the rate of 1 byte per pixel. It has 256 gray levels. In these experiments, the quantized transform coefficients are stored. The quantization threshold determines the amount of compression. High frequency coefficients are normally made zero in the above algorithm. But the low frequency coefficients take more than a byte for storage. Since the number of coefficients, which are quantized, is sufficiently large, the net storage is lesser than the storage of MR image at 1 byte per pixel. The results of algorithm 4.4.1 implemented on img.60 (test image) are presented below.

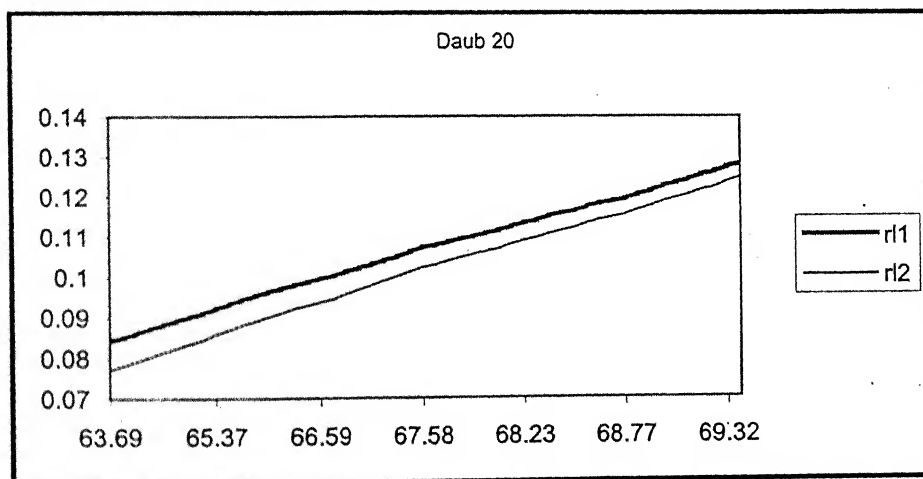
Img 60 using Daub4, Daub12, Daub20  
Percentage compression with Relative Errors r11,r12



Graph 4.4.1



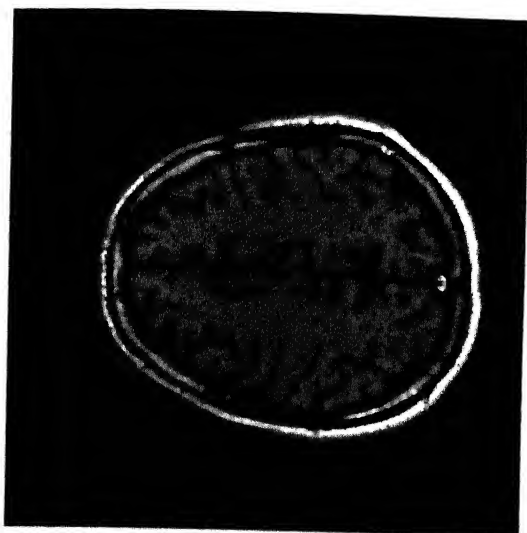
Graph 4.4.2



Graph 4.4.3

## COMPRESSION THROUGH WAVELETS (4.4.1)

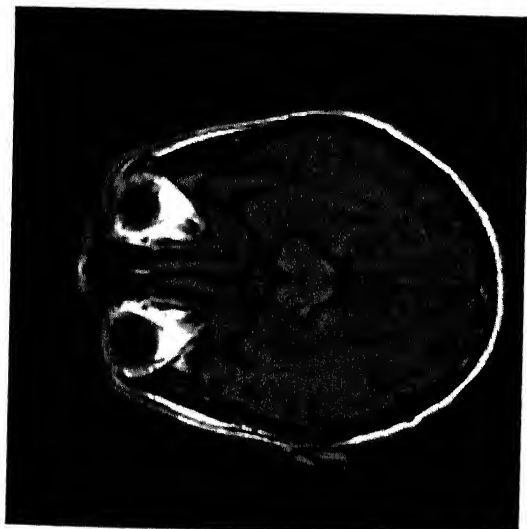
Images after decompression



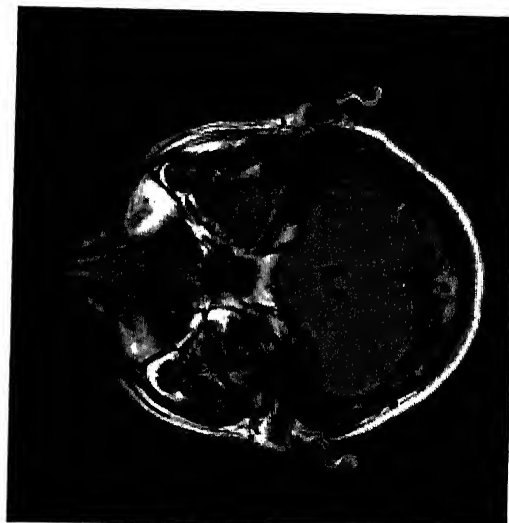
P.C : 56.69 % B.R : 3.46 bpp



P.C : 55.04 % B.R : 3.59 bpp



P.C.: 51.66 % B.R.: 3.87 bpp



P.C.: 46.53 % B.R.: 4.277 bpp

Image 4.4.1

**Compression through wavelets ( 4.4.1 )**  
**Images after decompression**

**Compression : 67 % Bitrate : 2.63 bpp**

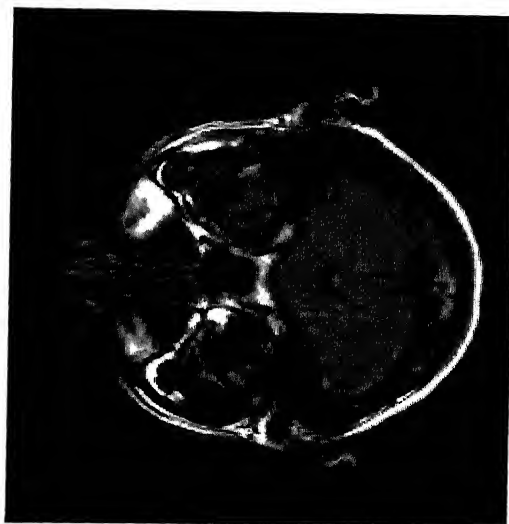


Image 4.4.2

From the Images 4.4.1-4.4.2 and Graph 4.4.2 we note that there is no visual loss of information in the decompressed images if the amount of compression done is not more than 56 % for the test images. Images compressed more than 60 % show up lot of deterioration in the architecture. Graphs 4.4.1-4.4.3 show that the performance of Daub 12 is better than Daub 4 and Daub 20 for the test images considered here. Comparing the images obtained after wavelet compression (Image 4.4.1-4.4.2) and images obtained after compressing through algorithm 4.2.1 (Image 4.2.1-4.2.5), we see that algorithm 4.2.1 implemented using Poisson process is better suited for MRI images. A comparison of the performance of standard compression techniques (Table 1.5.1) and that of algorithm 4.2.1 implemented using Poisson process (4.2.1) for the test images under consideration shows that the algorithm 4.2.1 performs better on the test images.

## REFERENCES

1. Abramson, J.R., and Krasner, B.H.: Data Compression in the PACS environment, *Proc. SPIE, Med. Imaging III*, vol.1091, pp. 42-49, 1989.
2. Antonini, M., Barlaud, M., Mathieu, P., Daubechies, I.: Image coding using wavelet transform, *IEEE Trans. Image Processing*, vol.1, pp. 205-220, Apr. 1992.
3. Averbuch, A., Lazar, D., and Israeli, M.: Image Compression Using Wavelet Transform and Multiresolution Decomposition, *IEEE Trans. on Image Processing*, vol. 5, 1, pp. 4-15, Jan. 1996.
4. Battle, G.: A block spin construction of wavelets, Part I Lemarie functions, *Comm. Math. Phys.*, vol.110, pp. 601-615, 1987.
5. Bergh, J. And Lofstorm, J.: *Interpolation Spaces*, Springer-Verlag, New York, 1976.
6. Bloch. F.: Nuclear Induction, *Phys. Rev.*, vol. 70, pp. 460-474, 1946.
7. Bransley, M.F.: *Fractals Everywhere*, Academic Press, 1992.
8. Brigham, E.O.: *The Fast Fourier Transform*, Englewood Cliffs, NJ: Prentice Hall, 1974.
9. Brown, C.W. and Shepherd, B.J.: *Graphics File Formats – Reference and Guide*, Manning Publications Co., 1995.
10. Butzer, P.L. and Berens, H.: *Semi-Groups of operators and approximation*, Springer-Verlag, New York, 1967.
11. Capton, J.: A probabilistic model for Run-Length coding of pictures, *IRE Trans. on Information Theory*, pp.157-163, 1959.
12. Chui, C.K.: *An Introduction to Wavelets*, Academic Press, 1992.
13. Cosman, P.C., Gray, R.M. and Olshen, R.A.: Evaluating Quality of Compressed Medical Images: SNR, Subjective Rating and Diagnostic Accuracy, *Proc. IEEE*, vol. 82, 6, pp. 919-932, June 1994.
14. Daubechies, I.: *Ten Lectures on Wavelets*, Philadelphia: SIAM, 1992.



15. Daubechies, I.: Orthonormal bases of compactly supported wavelets, *Comm. Pure Appl. Math.*, vol.41, pp. 909-996, 1988.
16. DeVore, R.A. and Popov, V.A.: Interpolation of Besov Spaces, *Trans. Amer. Math. Soc.*, vol.305, pp. 397-414, 1988.
17. DeVore, R.A., Jawerth, B. and Lucier, B.: Image compression through wavelet transform coding, *IEEE Trans. Inform. Theory*, vol.38, pp. 719-746, 1992.
18. DeVore, R.A., Jawerth, B. and Popov, V.A.: Compression of wavelet decompositions, *Amer. J. Math.*, vol.114, pp. 737-785, 1992.
19. Donoho, D.L., Vetterli, M., DeVore, R.A. and Daubechies, I.: Data Compression and Harmonic Analysis, *IEEE Trans. Inform. Theory*, vol.44, No. 6, pp. 2435-2475, Oct. 1998.
20. Gallager, R.G.: Variations on a theme by Huffman, *IEEE Trans. on Inform. Theory*, vol. IT-24, No. 6, pp. 668-674, Nov.1978.
21. Gersho, A., and Gray, R.M.: *Vector Quantization and signal compression*, Kulwer Academic Publishers, 1991.
22. Gitlin, J.N.: Teleradiology, *Radiologic Clin. North Amer.*, vol.24, pp. 55-68, Mar.1986.
23. Gray, R.M. and Neuhoff, D.L.: Quantization, *IEEE Trans. Inform. Theory*, vol. 44, No. 6, pp. 2325-2383, Oct. 1998.
24. Hamming, R.W.: *Digital Filters* 2<sup>nd</sup> ed., Englewood Cliffs, NJ: Prentice Hall, 1983.
25. Hardy, G.H., Littlewood, J.E., and Polya, G.: *Inequalities*, Cambridge University Press, 1959.
26. Healy, D.M., Jr. and Weaver, J.B.: Two Applications of Wavelet Transforms in Magnetic Resonance Imaging, *IEEE Trans. on Inform. Theory*, vol. 38, 2, Mar. 1992.
27. Hermen, G.T.: *Image Reconstruction from Projections. The fundamentals of Computerized Tomography*, Academic Press, New York, 1980.
28. Hildebrand, F.B.: *Introduction to Numerical Analysis*, Tata McGraw Hill, New Delhi, India, 1974.
29. Hinshaw, W.S., and Lent, A.H.: An Introduction to NMR Imaging: From the Bloch Equation to the Imaging Equation, *Proc. IEEE*, vol. 71, 3, pp. 338-350, Mar. 1983.

30. Huffman, D.A.: A method for the construction of minimum redundancy codes, *Proc. IRE*, vol. 40, pp. 1098-1101, 1952.
31. Hunter, R., and Robinson, A.H.: International Digital Facsimile coding standards, *IEEE Proc.*, vol.68, pp. 854-867, July 1980.
32. Jacquin, A.E.: Image coding based on a Fractal Theory of Iterated Contractive Image Transformations, *IEEE Trans. on Image Processing*, 1, pp. 8-30, Jan.1992.
33. Jacquin, A.E.: *A Fractal Theory of Iterated Markov operators with Applications to Digital Image coding*, Ph.D., thesis, Georgia Tech., 1989.
34. Jain, A.K.: *Fundamentals of Digital Image Processing*, Englewood Cliffs, NJ: Prentice Hall, 1989.
35. Jain, A.K.: Image data compression: a review, *Proc. IEEE*, vol. 69, pp. 349-389, Mar. 1981.
36. Jayant, N., Johnston, J., and Safranek, R.: Signal compression based on models of human perception, *Proc. IEEE*, vol.81, pp. 1385-1422, Oct. 1993.
37. Kaliprasad, V.S.N., Gupta, R.K., Rao, S.B., Kalyanaraman, R., Mahesh, Reddy, P.V.K., and Rathore, R.K.S.: Deblurring of MRI Images, *5<sup>th</sup> National Symposium on Magnetic Resonance*, IIP Dehradun, India, Feb. 23-26, 1999.
38. Kalyanaraman, R., Gupta, R.K., Mahesh, Rao, S.B., Kaliprasad, V.S.N., Reddy, P.V.K., and Rathore, R.K.S.: Storage of Raw data in Magnetic Resonance Imaging, *5<sup>th</sup> National Symposium on Magnetic Resonance*, IIP Dehradun, India, Feb. 23-26, 1999.
39. Kalyanaraman, R., Gupta, R.K., Mahesh, Rao, S.B., Kaliprasad, V.S.N., Reddy, P.V.K., and Rathore, R.K.S.: A New Compression Strategy for Magnetic Resonance Images, *Proc. Seventh Scientific Meeting of ISMRM at Philadelphia, USA*, May 24-28, 1999, in press.
40. Kolmogorov, A.N.: On the Shannon theory of information transmission in the case of continuous signals, *IEEE Trans. Inform. Theory*, vol. IT-2, pp.102-108, Sept. 1956.
41. Kuduvalli, G.R., and Rangayyan, R.M.: Performance Analysis of Reversible Image Compression Techniques for High-Resolution Digital Teleradiology, *IEEE Trans. on Med. Imag.*, vol.11, No.3, pp. 430-444, Sep.1992.
42. Kumar, A., Welti, D. and Ernst, R.R.: NMR Fourier Zeugmatography, *J. Magn. Reson.*, 18, pp. 69-82, 1975.
43. Kunth, D.E.: Dynamic Huffman Coding, *Journal of Algorithms*, 6: pp.163-180, 1985.

44. Langdon, G.G. and Rissanen, J.J., Compression of Black-white Images with Arithmetic coding, *IEEE Trans. on communications*, 29(6), pp. 858-867, 1981.
45. Lauterbur, P.C.: Image formation by induced local interactions: Examples employing nuclear magnetic resonance, *Nature*, vol. 242, pp.190-191, 1973.
46. Lorentz, G.G.: *Approximation of functions*, Holt, Rinehart and Winston, New York, 1966.
47. Mahesh, Gupta, R.K., Kalyanaraman, R., Rao S.B., Kaliprasad, V.S.N., Reddy, P.V.K., and Rathore, R.K.S.: Truncation Artifact Reduction in MRI with a New Class of Filters, *Proc. Seventh Scientific Meeting of ISMRM at Philadelphia, USA*, May 24-28, 1999, in press.
48. Mahesh, Gupta, R.K., Rao, S.B., Kaliprasad, V.S.N., Kalyanaraman, R., Reddy, P.V.K., and Rathore, R.K.S.: A Study of Resolution Using Classical Filters in MRI, *5<sup>th</sup> National Symposium on Magnetic Resonance*, IIP Dehradun, India, Feb. 23-26, 1999.
49. Mallat, S.G.: A Theory for Multiresolution Signal decomposition: The wavelet representation, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 674-693, July 1989.
50. Mansfield, P., and Morris, P.G.: *NMR Imaging in Biomedicine*, New York: Academic, 1982.
51. Mansfield, P. and Pykett, I.L.: Biological and medical imaging by NMR, *J. Magn. Reson.*, vol. 29, 2, pp. 355-373, Feb. 1978.
52. Meyer, Y.: *Wavelets and Operators*, Cambridge Univ. Press, 1992.
53. Natanson, I.P.: *Constructive function theory*, vol. I, Frederide Unger Publ., Co., New York, 1964.
54. Nijim, Y.W., Stearns, S.D., and Mikhael, W.B.: Differentiation applied to lossless compression of Medical Images, *IEEE Trans. on Med. Imag.*, vol.15, No.4, pp. 555-559, Aug.1996.
55. Partain C.L., Price, R.R., Patton, J.A., Kulkarni, M.V. and James, A.E. Jr.: *Magnetic Resonance Imaging*, vol. I, Clinical Principles, W.B.Saunders Company, Harcourt Brace Jovanovich, Inc., Philadelphia, 1988.
56. Partain C.L., Price, R.R., Patton, J.A., Kulkarni, M.V. and James, A.E. Jr.: *Magnetic Resonance Imaging*, vol. II, Physical Principles and Instrumentation, W.B.Saunders Company, Harcourt Brace Jovanovich, Inc., Philadelphia, 1988.

57. Peteraj, A.W.H.: *Global  $L_p$  approximation by linear combination of certain positive operators*, Ph.D. Thesis, I.I.T. Kanpur, 1982.
58. Peters, T.M. and Williams, J.: *The Fourier Transform in Biomedical Engineering*, Birkhauser, Boston, 1998.
59. Ramaswamy, A., and Mikhael, W.B.: A Mixed Transform Approach for Efficient Compression of Medical Images, *IEEE Trans. on Med. Imag.*, vol.15, 3, pp. 343-352, June 1996.
60. Rao, S.B., Gupta, R.K., Kaliprasad, V.S.N., Mahesh, Kalyanaraman, R., Reddy, P.V.K., and Rathore, R.K.S.: Tissue Parameters Estimation in MR Imaging, 5<sup>th</sup> *National Symposium on Magnetic Resonance*, IIP Dehradun, India, Feb. 23-26, 1999.
61. Rathore, R.K.S.: On a sequence of linear trigonometric polynomial operators, *SIAM J. Math. Anal.* vol. 5, No.6, pp. 908-917, 1974.
62. Rathore, R.K.S.: Optimal inputs for Approximate Linear Systems in Hilbert Spaces, *Jl. of Math. Analysis and Appln.*, 188, pp. 398-409, 1994.
63. Ross, P., Viergever, M.A., Van Dijke, M.C.A. and Peters, J.H.: Reversible Intraframe Compression of Medical Images, *IEEE Trans. on Med. Imag.*, vol.7, 4, pp. 328-336, Dec. 1988.
64. Rudin, W.: *Real and complex analysis*, Tata McGraw Hill, India, 1979.
65. Sayood, K.: *Introduction to Data Compression*, San Francisco, CA: Morgan Kaufmann, 1996.
66. Sebastiani, G. and Barone, P.: Mathematical principles of basic magnetic resonance imaging in medicine, *Signal Processing*, 25, pp. 227-250, 1991.
67. Shannon, C.E.: Prediction and Entropy of printed English, *Bell Sys. Tech. Jl.*, vol.30, pp. 50-64, Jan.1951.
68. Shannon, C.E.: The mathematical theory of communication, part I and II, *Bell Syst. Tech. Jl.*, vol.27, pp. 379-423, and 623-686, 1948.
69. Startk, D.D. and Bradley, W.G.: *Magnetic Resonance Imaging*, Mosby-year Book, St.Louis, Missouri, 1992.
70. Stromberg, J.O.: A modified Harr system and higher order spline systems, in *Conf. in Harmonic Analysis in Honor of Antoni Zygmund*, vol. II, pp. 475-493.

71. Timan, A.F.: *Theory of approximation of functions of a real variable*, Hindustan Publishing Corporation, Delhi, India, 1966.
72. Vaidyanathan, P.P.: Quadrature Mirror filter banks, M-B Extension and perfect reconstruction Techniques, *IEEE ASSP Msg.*, vol.4, pp. 4-20, July 1984.
73. Wallace, G.K.: The JPEG still picture compression standard, *Commun. ACM*, vol.34, no.4, pp. 30-45, Apr.1991.
74. Welch, T.A.: A Technique for High-Performance Data Compression, *IEEE computer*, pp. 8-19, June 1984.
75. Wojtaszczyk, K.P.: *A Mathematical Introduction to Wavelets*, Cambridge Univ. Press, 1997.
76. Y.Fisher, editor. : *Fractal Image Compression Theory and Application*, New York, Springer-Verlag, 1995.
77. Ziv, J. and Lempel, A.: A Universal Algorithm for Data Compression, *IEEE Trans. on Inform. Theory*, IT-23 (3), pp. 337-343, May 1977.
78. Ziv.J and Lempel.A.: Comprssion of Individual Sequences via variable Rate Coding. *IEEE Trans. on Inform. Theory*, IT-24 (5), pp. 530-536, September 1978.
79. Zygmund, A.: *Trigonometric Series*, vol. I, Cambridge University Press, London-New York, 1968.

**A** 139687



A139687